

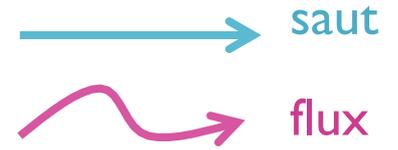
# **Modélisation hétérogène – Systèmes hybrides**

# Modélisation hétérogène – Systèmes hybrides

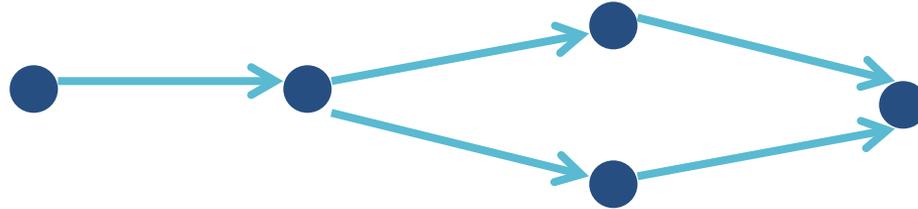
---

1. **Système hybride: introduction**
  
2. **Système Temps-Réel: automate temporisé**
  - ▶ Modélisation
  - ▶ Vérification
  
3. **Système Hybride: automate hybride**
  - ▶ Modélisation
  - ▶ Vérification

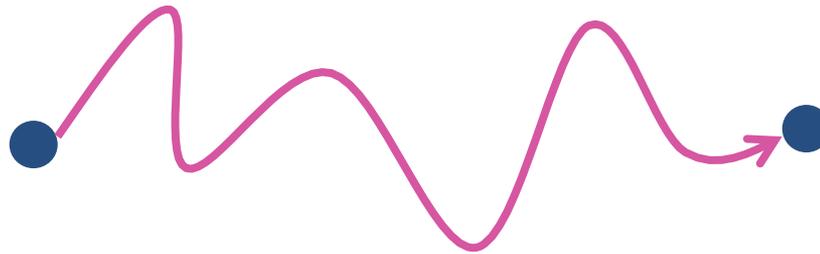
# Systeme hybride



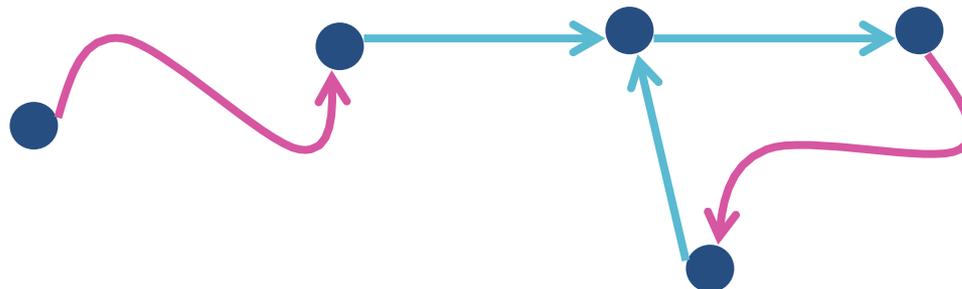
- ▶ Dynamique **discrète**  
(machine à états finis)



- ▶ Dynamique **continue**  
(équation différentielle)



- ▶ Systeme **hybride**



# Exemple : Thermostat

---

- ▶ Variables
  - ▶ Température
  - ▶ Mode

# Exemple : Thermostat

---

- ▶ Variables
  - ▶ Température : **variable continue**  $T$  à valeurs réelles
  - ▶ Mode : **variable discrète**  $M$  à valeurs dans  $\{\text{ON}, \text{OFF}\}$

# Exemple :

## Thermostat

---

- ▶ Variables
  - ▶ Température : **variable continue**  $T$  à valeurs réelles
  - ▶ Mode : **variable discrète**  $M$  à valeurs dans  $\{ON, OFF\}$
- ▶ **Flux** dans chacun des modes
  - ▶ Mode ON :  $\dot{T} = K \times (100 - T)$   
*invariants* :  $(M = ON) \wedge (T < 82^\circ)$
  - ▶ Mode OFF :  $\dot{T} = -K \times T$   
*invariants* :  $(M = OFF) \wedge (T > 68^\circ)$

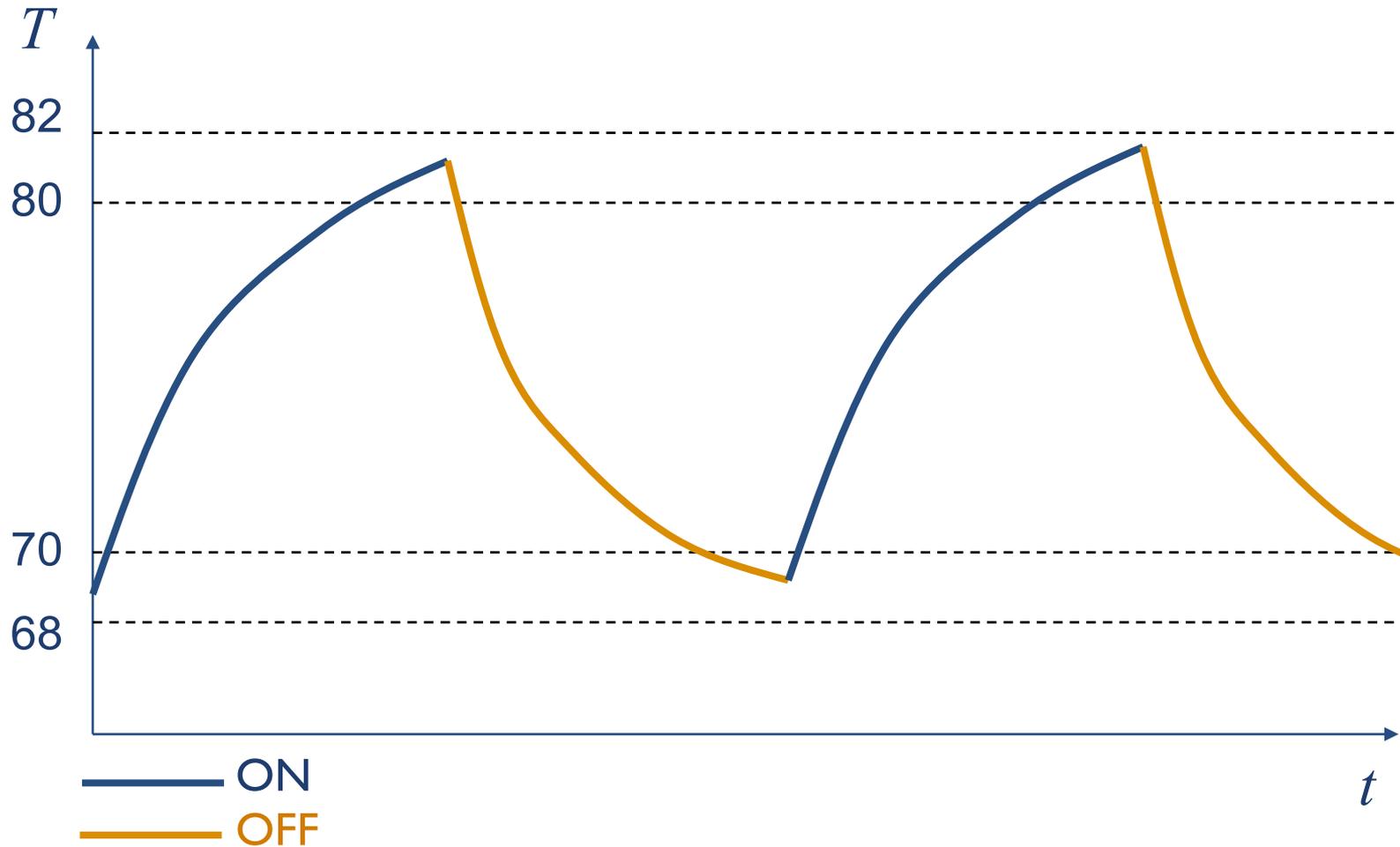
# Exemple :

## Thermostat

---

- ▶ Variables
  - ▶ Température : **variable continue**  $T$  à valeurs réelles
  - ▶ Mode : **variable discrète**  $M$  à valeurs dans  $\{ON, OFF\}$
- ▶ **Flux** dans chacun des modes
  - ▶ Mode ON :  $\dot{T} = K \times (100 - T)$   
*invariants* :  $(M = ON) \wedge (T < 82^\circ)$
  - ▶ Mode OFF :  $\dot{T} = -K \times T$   
*invariants* :  $(M = OFF) \wedge (T > 68^\circ)$
- ▶ **Sauts** entre les modes (instantanés)
  - ▶ Si  $(M = ON) \wedge (T \geq 80^\circ)$  :      ON  $\longrightarrow$  OFF
  - ▶ Si  $(M = OFF) \wedge (T \leq 70^\circ)$  :      OFF  $\longrightarrow$  ON

# Exemple : Dynamique du thermostat



# Autres exemples de systèmes hybrides

- ▶ Systèmes *naturels* présentant différentes phases de comportement
  - ▶ Balle rebondissante
  - ▶ Croissance de cellules



- ▶ Systèmes avec contrôle numérique
  - ▶ Thermostat
  - ▶ Contrôleur d'un système de production d'énergie
  - ▶ Régulateur de vitesse de voiture
  - ▶ Pilote automatique d'avion



# Modélisation hétérogène – Systèmes hybrides

---

1. Système hybride: introduction
  
2. **Système Temps-Réel: automate temporisé**
  - ▶ **Modélisation**
  - ▶ Vérification
  
3. Système Hybride: automate hybride
  - ▶ Modélisation
  - ▶ Vérification

# Systeme temps-réel

---

Radar System

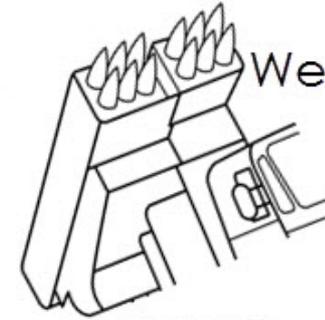


**REAL-TIME SYSTEM**

Incoming Missile 



Command &  
Decision System

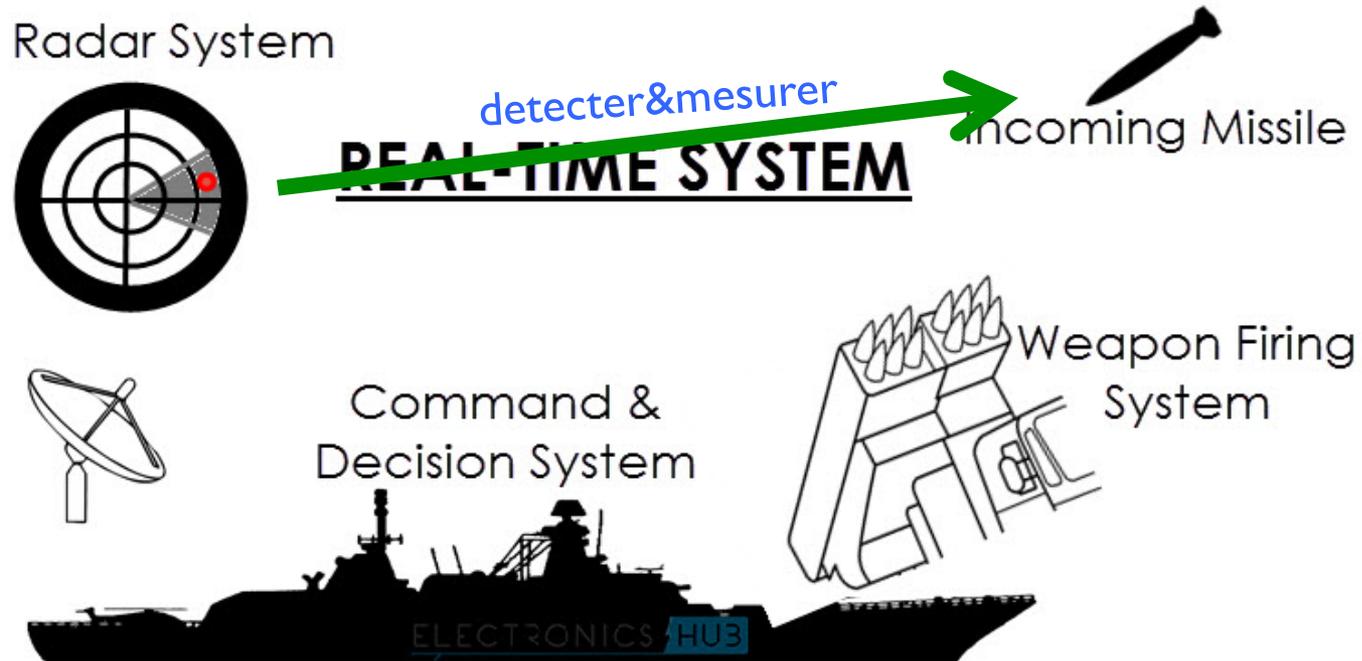


Weapon Firing  
System



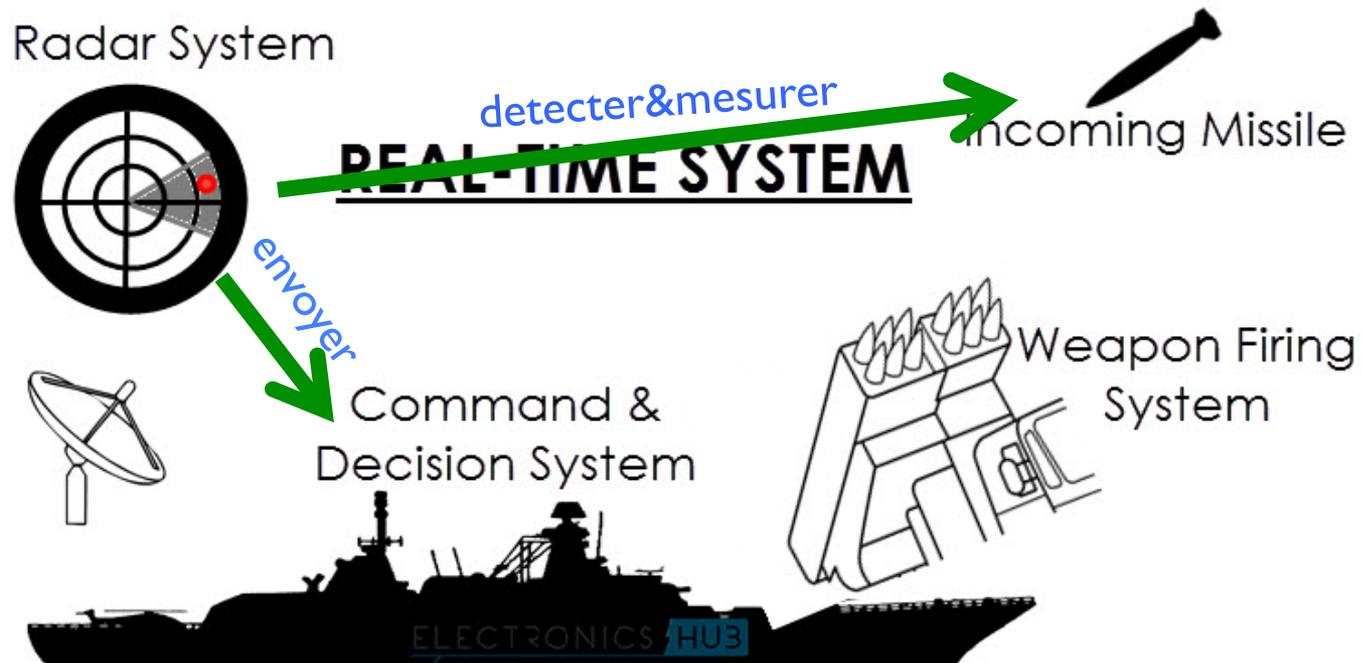
# Systeme temps-réel

---

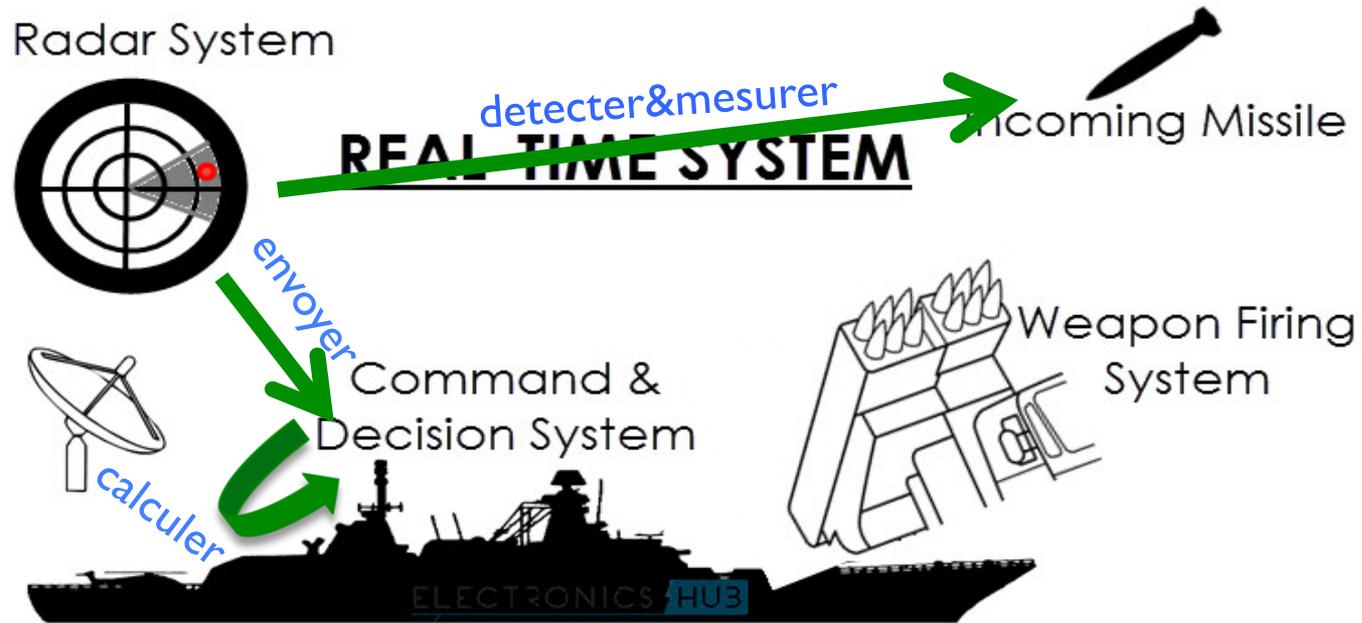


# Systeme temps-réel

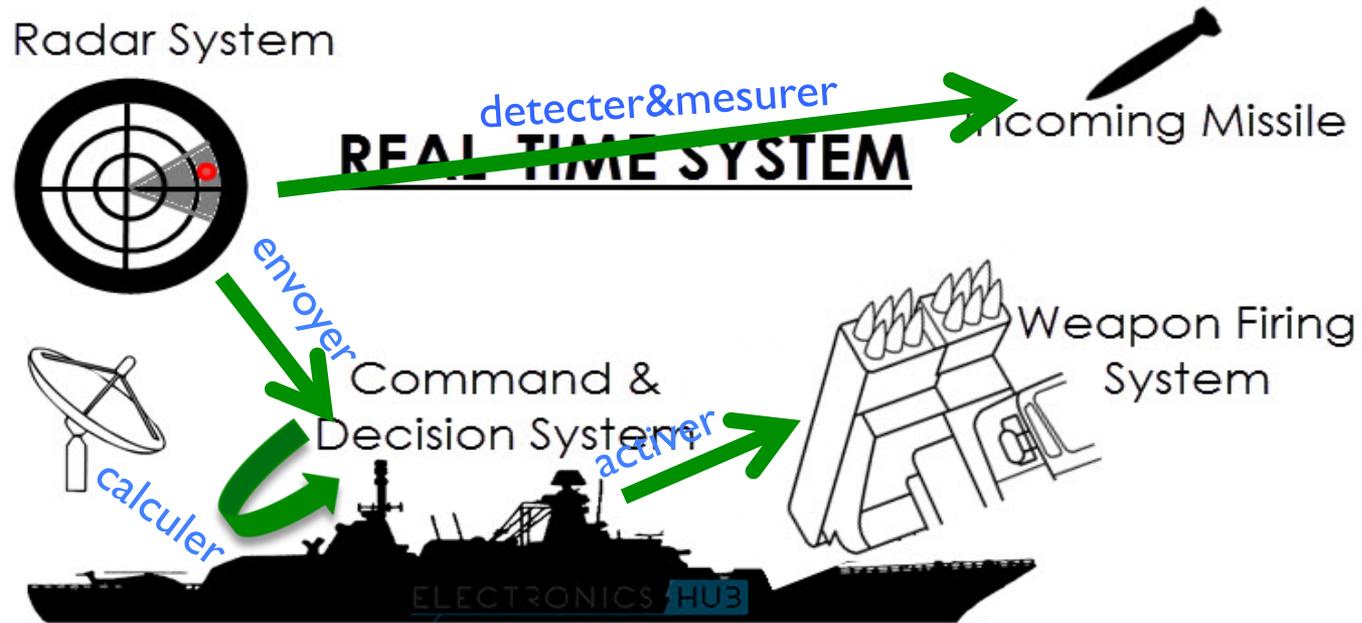
---



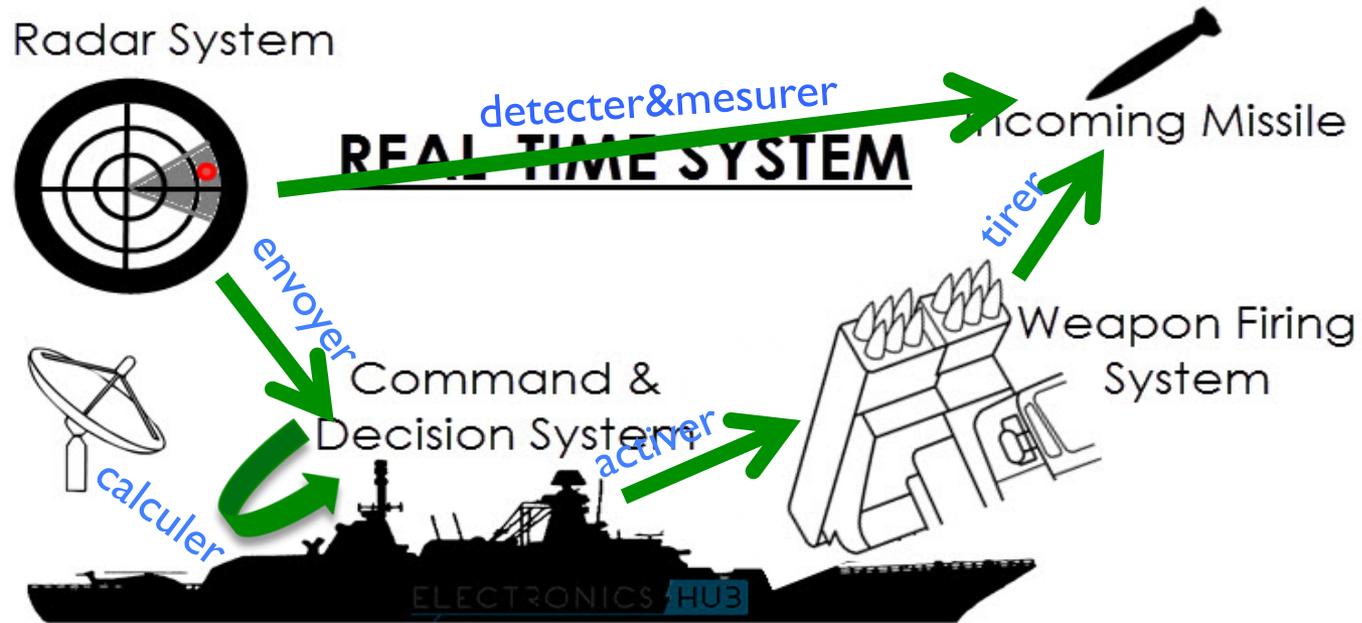
# Systeme temps-réel



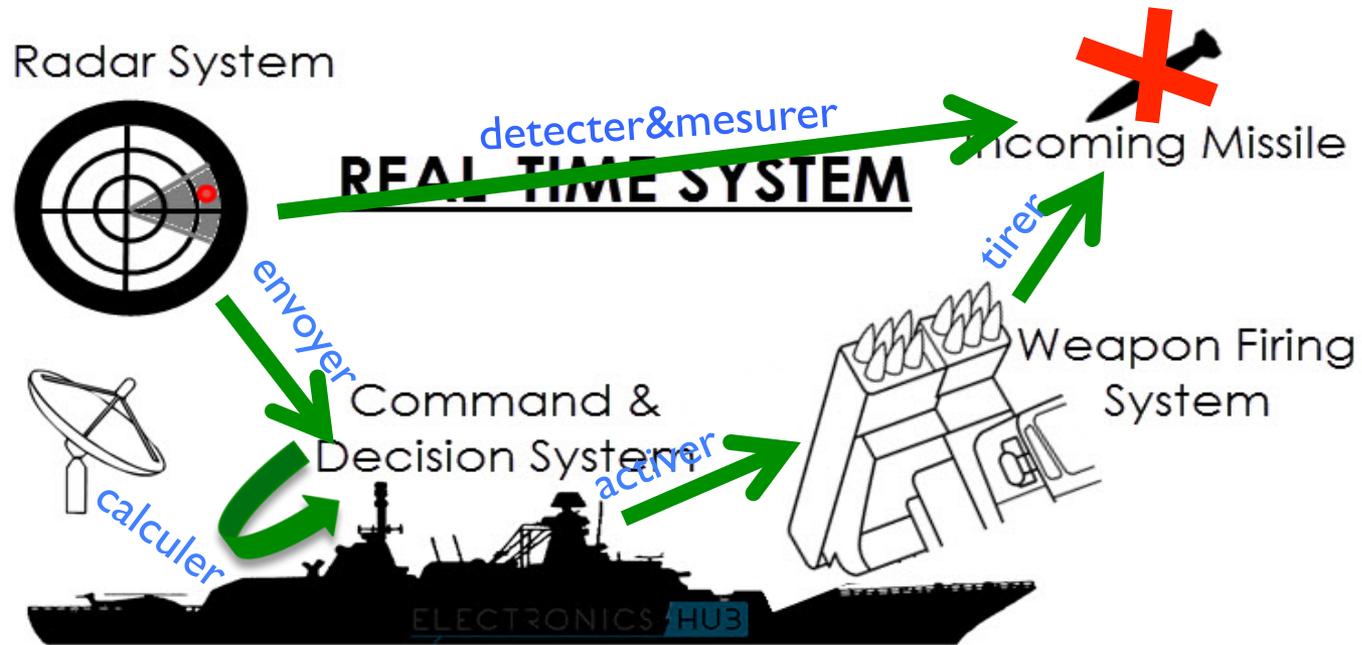
# Systeme temps-réel



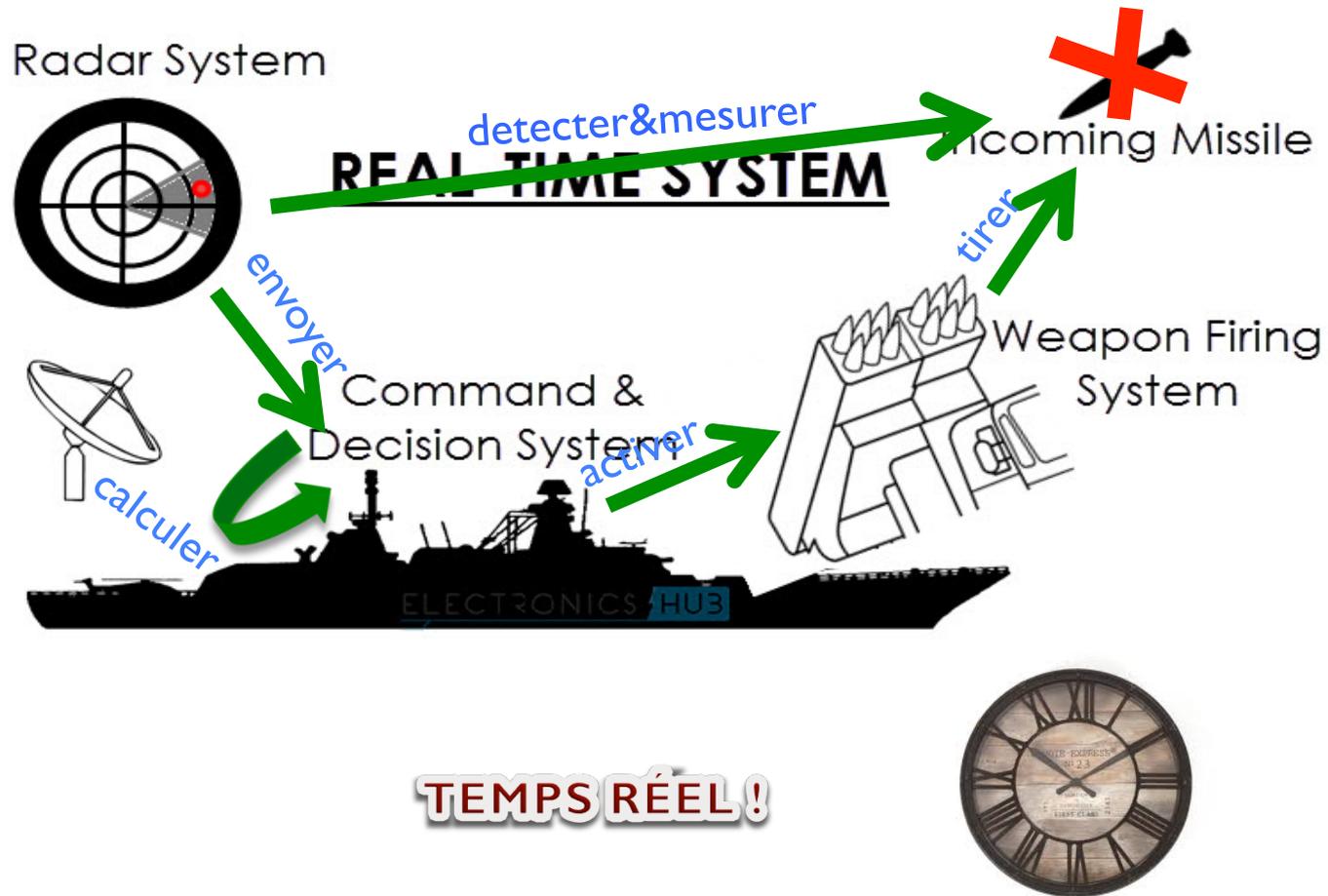
# Systeme temps-réel



# Systeme temps-réel



# Systeme temps-réel



# Exemple: Patriot (Guerre du Golfe)

---

25 février 1991, guerre du Golfe  
28 soldats morts et 100 blessés



# Exemple: Patriot (Guerre du Golfe)

---

25 février 1991, guerre du Golfe  
28 soldats morts et 100 blessés

Problème: erreur d'horloge  
Horloge interne s'augmente par 1/10 chaque 1/10 s

Horloge enregistré dans un registre de 24-bit  
 $1/10 - \langle 1/10 \rangle_{24\text{bit}} \approx 10^{-7}$



# Exemple: Patriot (Guerre du Golfe)

---



25 février 1991, guerre du Golfe  
28 soldats morts et 100 blessés

Problème: erreur d'horloge  
Horloge interne s'augmente par 1/10 chaque 1/10 s

Horloge enregistré dans un registre de 24-bit  
 $1/10 - \langle 1/10 \rangle_{24\text{bit}} \approx 10^{-7}$

**Après 100 heures, l'écart total était de 0,34 seconde.  
Le missile n'a pas pu être détruit**

# Systeme temps-réel

---

Définition: systèmes dont la correction ne dépend pas seulement des valeurs des résultats produits mais également des **délais** dans lesquels les résultats sont produits.

# Systeme temps-réel

---

Définition: systèmes dont la correction ne dépend pas seulement des valeurs des résultats produits mais également des **délais** dans lesquels les résultats sont produits.

Un système temps réel n'est pas un système ~~qui va vite~~ mais un système qui **satisfait à des contraintes temporelles**.

# Systeme temps-reel

---

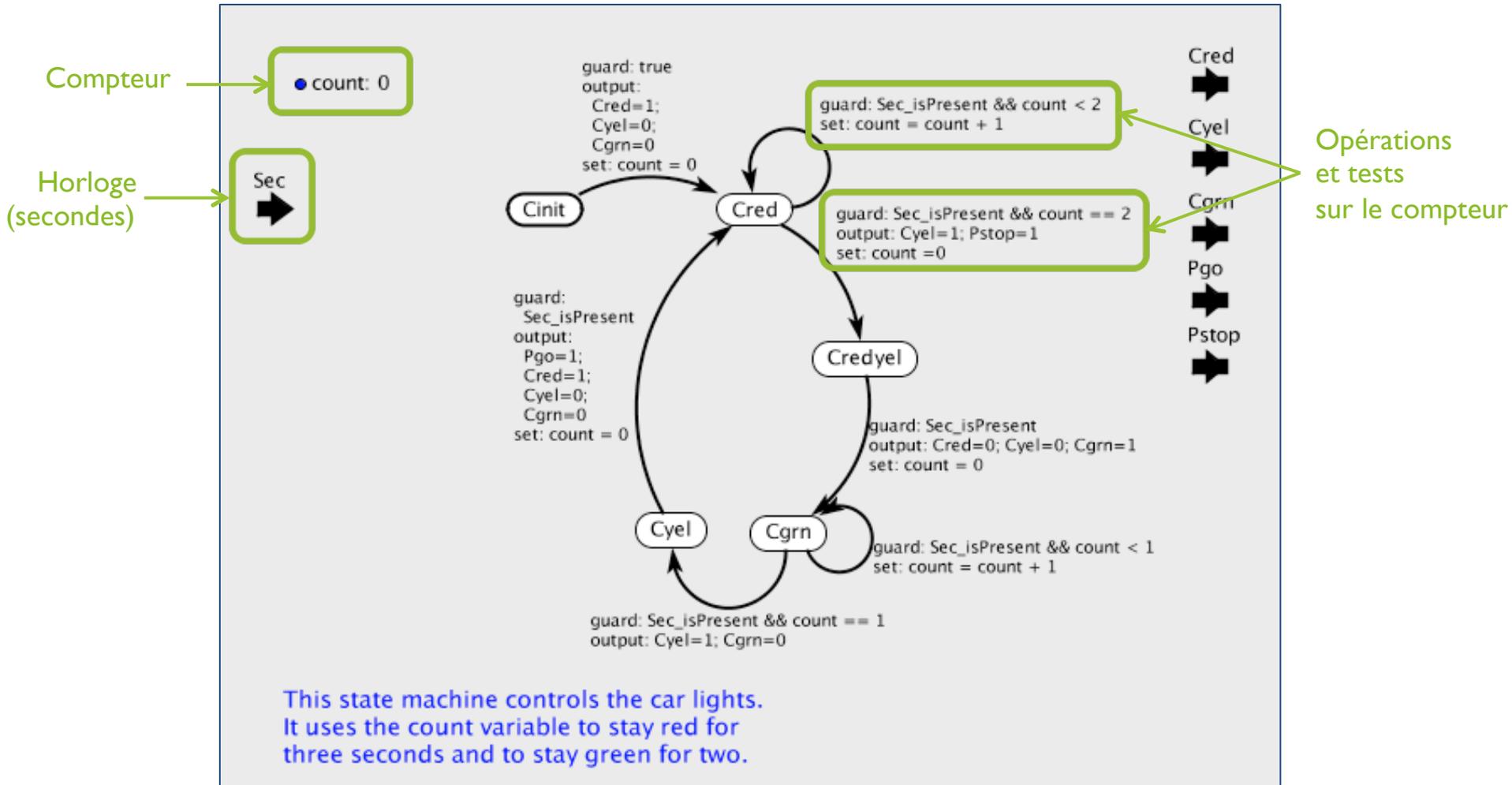
Définition: systèmes dont la correction ne dépend pas seulement des valeurs des résultats produits mais également des **délais** dans lesquels les résultats sont produits.

Un système temps réel n'est pas un système ~~qui va vite~~ mais un système qui **satisfait à des contraintes temporelles**.

Exemples de grandeur:

- La milliseconde pour les systèmes radar.
- La seconde pour les systèmes de visualisation humain.
- Quelques heures pour le contrôle de production impliquant des réaction chimiques
- 24 heures pour les prévisions météo
- Plusieurs mois ou année pour les systèmes de navigation de sonde spatiale.

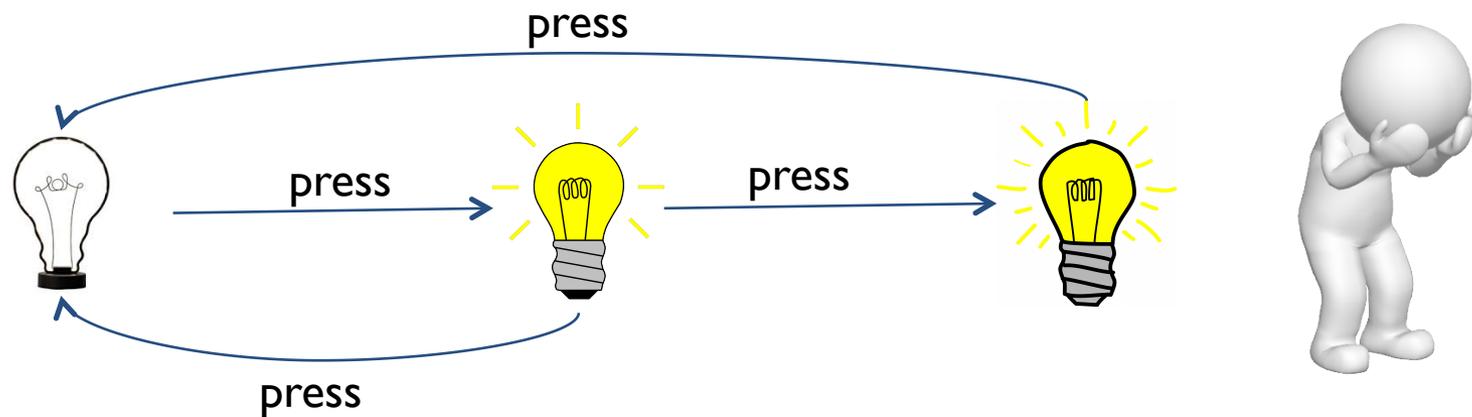
# Comment?



# Comment?

---

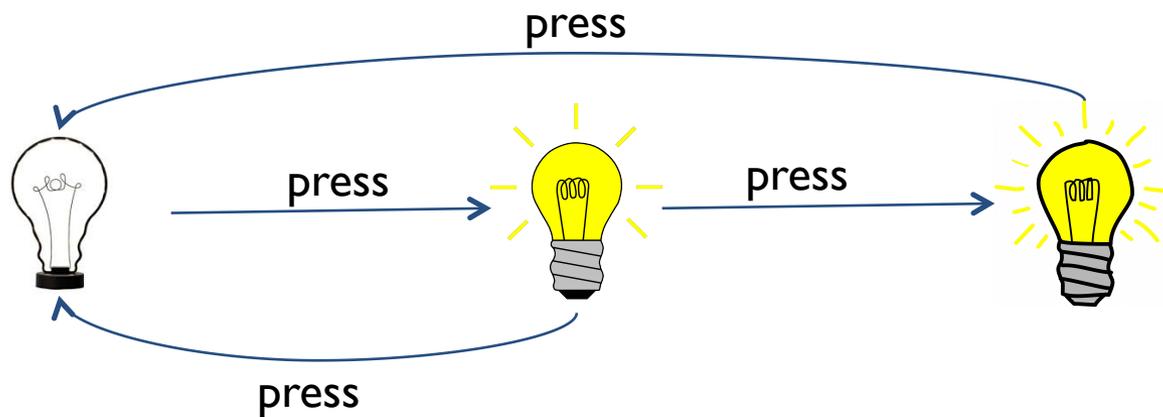
Si l'on appuie **deux** fois rapidement sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.



# Comment?

---

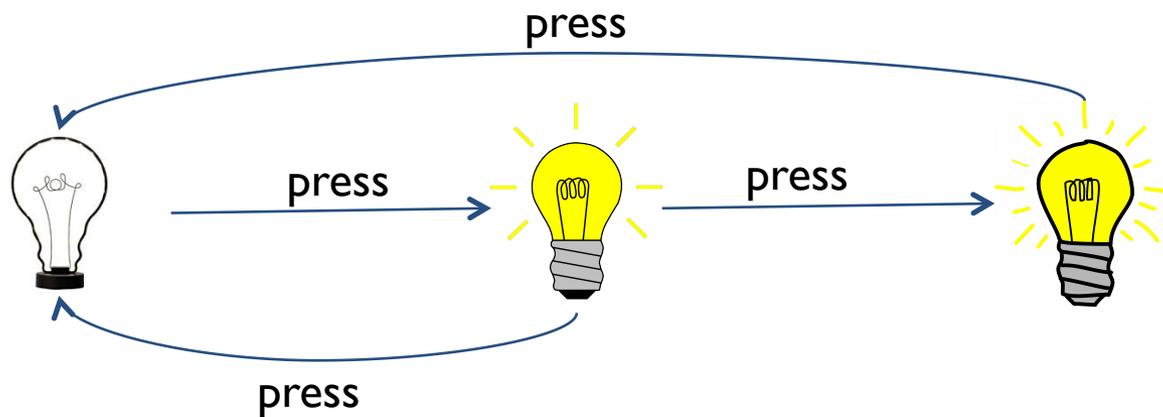
Si l'on appuie **deux** fois **rapidement** sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.



# Comment?

---

Si l'on appuie **deux** fois **rapidement** sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.

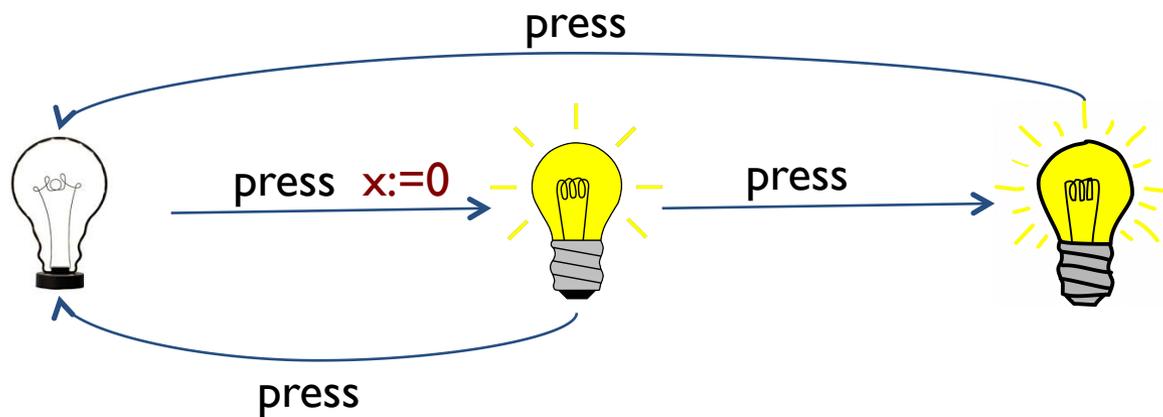


Pourquoi pas horloge ?

# Comment?

---

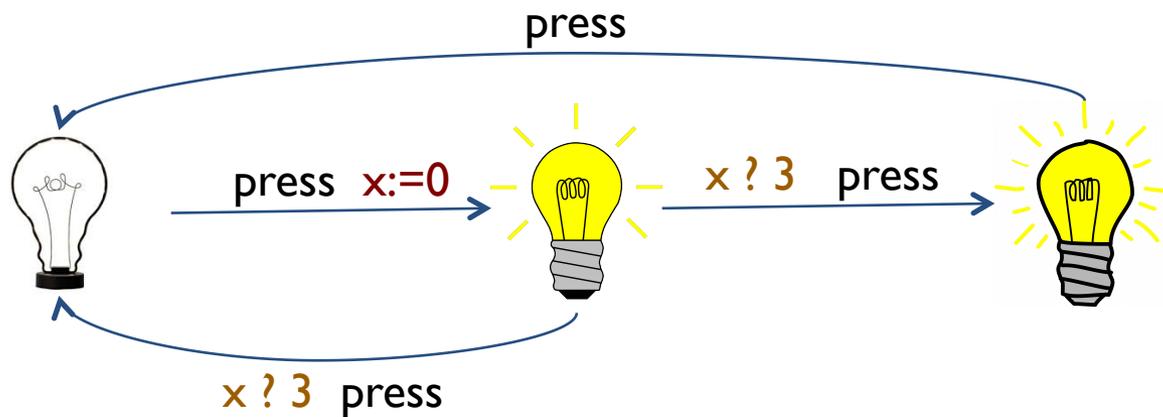
Si l'on appuie **deux** fois **rapidement** sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.



Pourquoi pas horloge ?

# Comment?

Si l'on appuie **deux** fois **dont la durée est inférieur ou égale à 3'** sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.

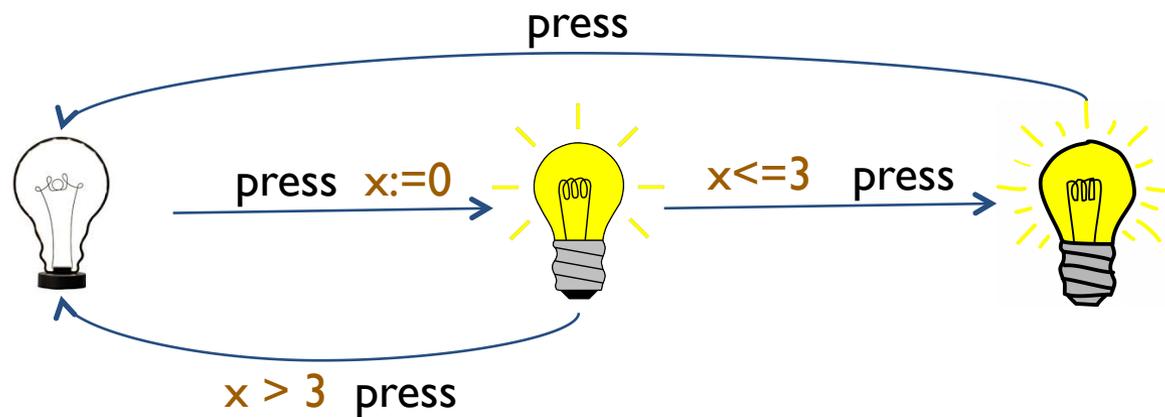


Pourquoi pas horloge ?

# Comment?

---

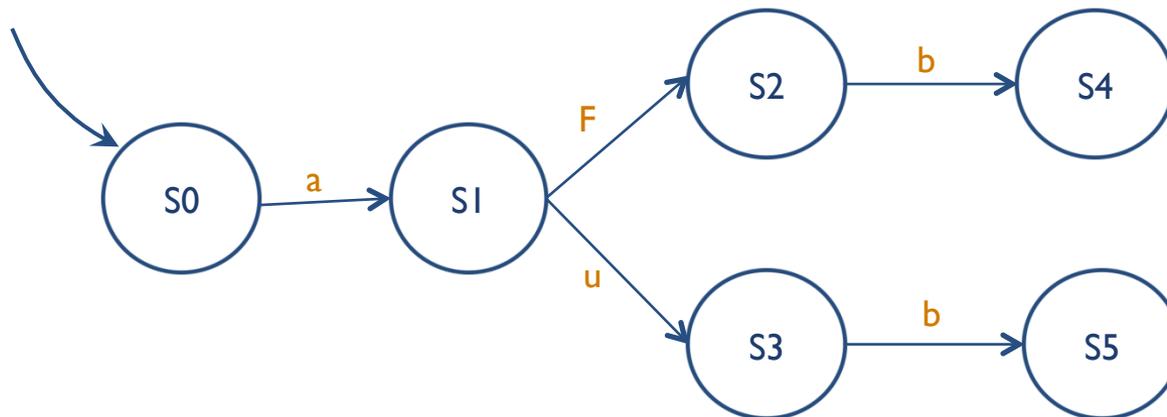
Si l'on appuie **deux** fois **dont la durée est inférieur ou égale à 3'** sur le bouton, alors la lumière sera plus claire, sinon la lumière sera éteinte.



# Pourquoi temps?

---

- ▶ Certaines propriétés dépendent de l'aspect temporel
- ▶ Exemple: Diagnosticabilité
  - ▶ Garantir le diagnostic d'une faute
  - ▶ Deux traces: une avec la faute + les mêmes observations

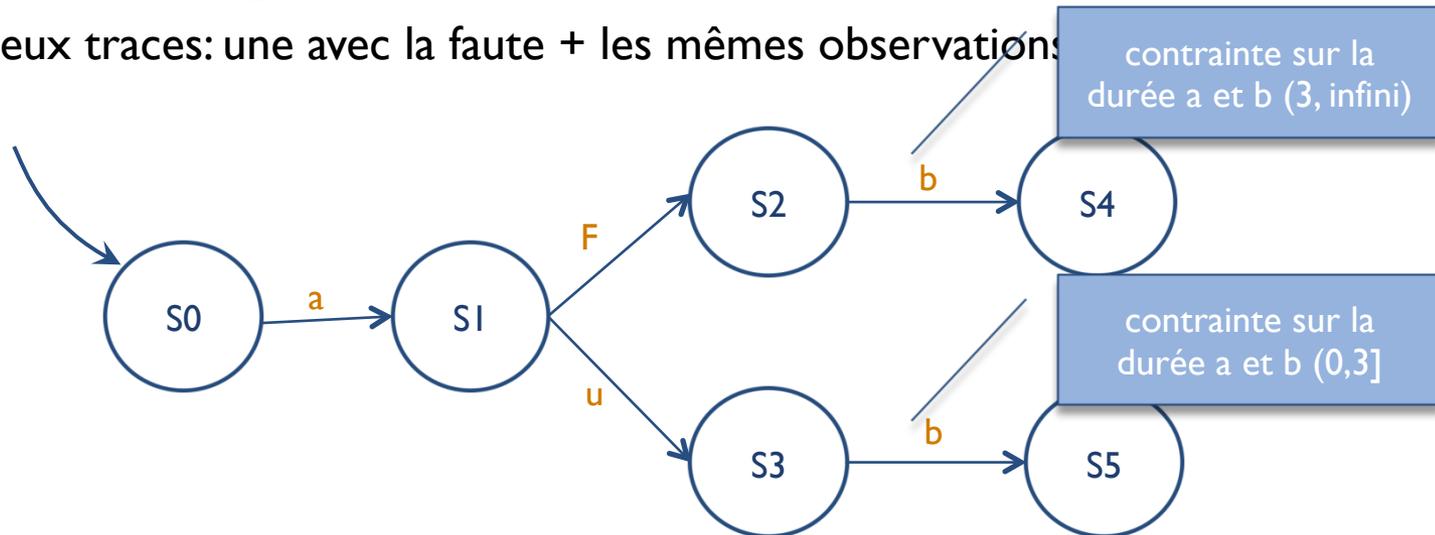


Deux traces (a.**F**.b et a.u.b) avec même observations: a.b

**Non diagnosticable**

# Pourquoi temps?

- ▶ Certaines propriétés dépendent de l'aspect temporel
- ▶ Exemple: Diagnosticabilité
  - ▶ Garantir le diagnostic d'une faute
  - ▶ Deux traces: une avec la faute + les mêmes observations

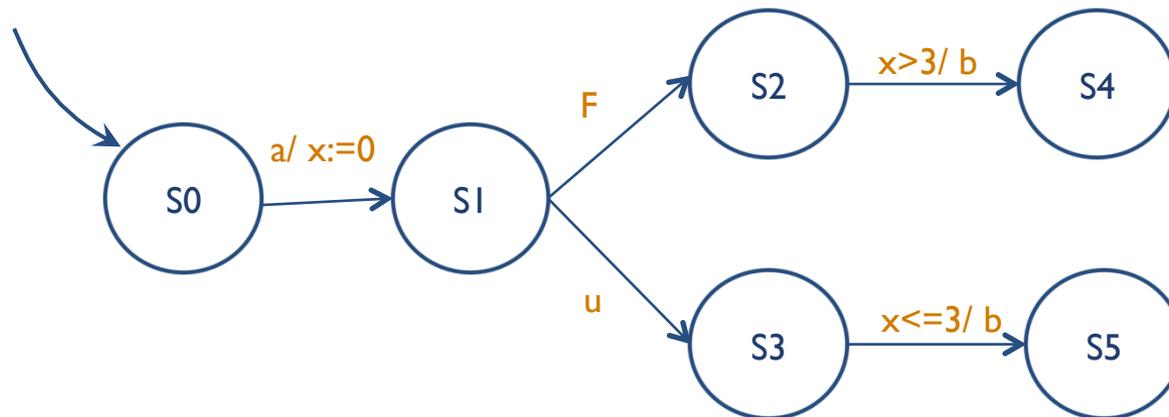


Deux trace (a.**F**.b et a.u.b) avec même observations: a.b ?

**Non diagnosticable ou diagnosticable?**

# Pourquoi temps?

- ▶ Certaines propriétés dépendent de l'aspect temporel
- ▶ Exemple: Diagnosticabilité
  - ▶ Garantir le diagnostic d'une faute
  - ▶ Deux traces: une avec la faute + les mêmes observations



Deux traces (a.**F**.b et a.u.b) avec même observations: a.b?

La relation temporelle entre 'a' et 'b' permet de diagnostiquer

**Diagnosticable**

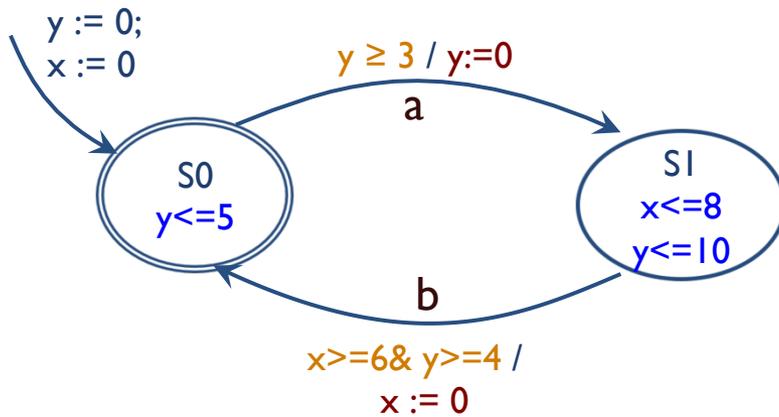
# Présentation

---

- ▶ **Automate temporisé** = automate fini + horloges
  - ▶ **Horloges** qui mesurent le passage du temps
    - ▶ **Avancent simultanément**, avec une dérivée constante
    - ▶ Peuvent être **réinitialisées** indépendamment
    - ▶ Valeurs réelles, positives ou nulles
  - ▶ Un nombre fini de **places (locations)**
  - ▶ **Transitions** sont instantanées
  - ▶ Le temps peut s'écouler dans chaque place
  - ▶ Les transitions sont conditionnées par **gardes**, peuvent remettre horloges à 0
  - ▶ **Invariant** peut être associé à chaque place
- ▶ Automate temporisé = **forme la plus « simple » d'un automate hybride**
  - ▶ Seul flot continu = passage du temps

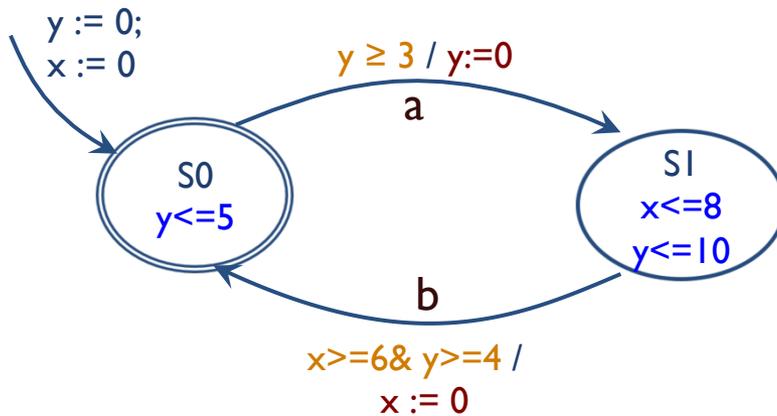
# Exemple

---



# Exemple

---



Horloges:  $x, y$

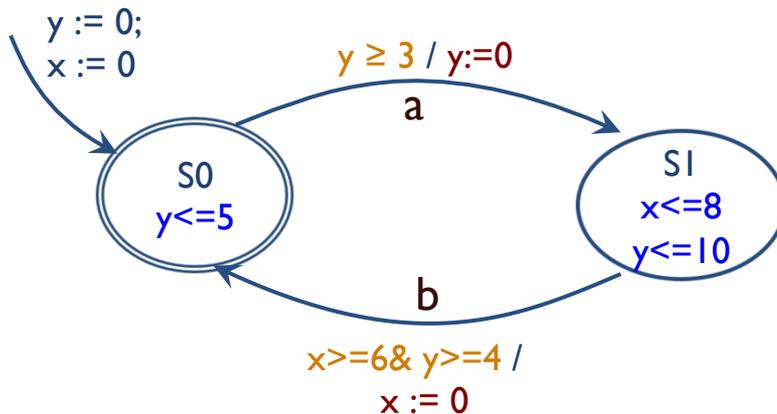
Places: S0 (initiale), S1

Invariants:

Transition (5 composants):

- état source
- garde
- événement
- réinitialisation
- état destination

# Exemple



Horloges:  $x, y$

Places: S0 (initiale), S1

Invariants:  $y \leq 5$  (S0)

$x \leq 8 \& y \leq 10$  (S1)

Transition (5 composants):

- état source
- garde
- événement
- réinitialisation
- état destination

T1: S0,  $y \geq 3$ , a,  $y := 0$ , S1

T2: S1,  $y \geq 4 \& x \geq 6$ , b,  $x := 0$ , S0

# Contraintes d'horloges (garde, invariant)

---

- Une horloge  $x$  est une variable à valeur dans  $\mathbb{R}^{\geq 0}$ , l'ensemble des réels positifs ou nuls.
- Soit  $X$  un ensemble fini d'horloges. L'ensemble  $C(X)$  des contraintes d'horloges est défini par la grammaire

**True |  $x < c$  |  $x \leq c$  |  $x > c$  |  $x \geq c$  |  $x = c$  |  $\Phi 1$  |  $\wedge$  |  $\Phi 2$**

où  $x \in X$  est une horloge,  $c \in \mathbb{Q}$  est un rationnel,  $\Phi 1$  et  $\Phi 2$  sont des contraintes d'horloges.

# Définition formelle (syntaxe)

---

Un automate temporisé est  $A = (\Sigma, S, S_0, X, I, T)$

Avec

- $\Sigma$  un alphabet fini.
- $S$  un ensemble fini de places (locations).
- $S_0 \subset S$  un ensemble de places initiales.
- $X$  un ensemble fini de variables (horloges).
- $I : S \rightarrow C(X)$  des invariants de places.
- $T \subset S \times \Sigma \times C(X) \times 2^X \times S$  un ensemble de transitions d'actions.

Chaque  $e = \langle s, a, \varphi, \lambda, s' \rangle \in T$  correspond à une transition entre la place  $s$  et la place  $s'$ , gardée par la contrainte  $\varphi$ , étiquetée par la lettre  $a$ , et qui remet les variables de  $\lambda \subset X$  à 0.

# Modèle-Interprétation (sémantique)

---

Un automate temporisé  $A = (\Sigma, S, S_0, X, I, T)$  s'interprète comme le système de transition temporisé avec un ensemble d'états (*infini*)  $ST(A) = (\Sigma, Q, Q_0, R)$  tel que

- $Q$ , l'ensemble des états, est constitué des couples  $(s, v)$ .  
 $s \in S$  donne la **place**, et  $v$  est une **valuation** des horloges.
- $Q_0 \subset Q$ , l'ensemble des **états initiaux** est constitué des couples  $(s, v)$  avec  $s \in S_0$  et  $v(x) = 0$  pour tout  $x \in X$ .
- $R$ , la **relation de transition** est décrite dans les transparents suivants.

# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

Transitions de **temps**: pour  $d \in \mathbb{R}^{\geq 0}$ , on note  $(s, v) \xrightarrow{d} (s, v')$  si  $v' = v + d$  et pour tout  $0 \leq e \leq d$ ,  $v + e$  vérifie la contrainte  $l(s)$ .

# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

Transitions de **temps**: pour  $d \in \mathbb{R}^{\geq 0}$ , on note  $(s, v) \xrightarrow{d} (s, v')$  si  $v' = v + d$  et pour tout  $0 \leq e \leq d$ ,  $v + e$  vérifie la contrainte  $I(s)$ .

Transitions d'**action**: pour  $a \in \Sigma$ , on note  $(s, v) \xrightarrow{a} (s', v')$  s'il existe  $\langle s, a, \varphi, \lambda, s' \rangle \in T$  tel que:  $v$  satisfait  $\varphi$ ;  $v' = v[\lambda := 0]$ ;  $v'$  satisfait  $I(s')$ .

# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

Transitions de **temps**: pour  $d \in \mathbb{R}^{\geq 0}$ , on note  $(s, v) \xrightarrow{d} (s, v')$  si  $v' = v + d$  et pour tout  $0 \leq e \leq d$ ,  $v + e$  vérifie la contrainte  $I(s)$ .

Transitions d'**action**: pour  $a \in \Sigma$ , on note  $(s, v) \xrightarrow{a} (s', v')$  s'il existe  $\langle s, a, \varphi, \lambda, s' \rangle \in T$  tel que:  $v$  satisfait  $\varphi$ ;  $v' = v[\lambda := 0]$ ;  $v'$  satisfait  $I(s')$ .



# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

Transitions de **temps**: pour  $d \in \mathbb{R}^{\geq 0}$ , on note  $(s, v) \xrightarrow{d} (s, v')$  si  $v' = v + d$  et pour tout  $0 \leq e \leq d$ ,  $v + e$  vérifie la contrainte  $I(s)$ .

Transitions d'**action**: pour  $a \in \Sigma$ , on note  $(s, v) \xrightarrow{a} (s', v')$  s'il existe  $\langle s, a, \varphi, \lambda, s' \rangle \in T$  tel que:  $v$  satisfait  $\varphi$ ;  $v' = v[\lambda := 0]$ ;  $v'$  satisfait  $I(s')$ .



**XOR**



# Relation de Transition

---

Deux types de transitions:

1. des transitions de temps,
2. des transitions d'action.

Transitions de **temps**: pour  $d \in \mathbb{R}^{\geq 0}$ , on note  $(s, v) \xrightarrow{d} (s, v')$  si  $v' = v + d$  et pour tout  $0 \leq e \leq d$ ,  $v + e$  vérifie la contrainte  $I(s)$ .

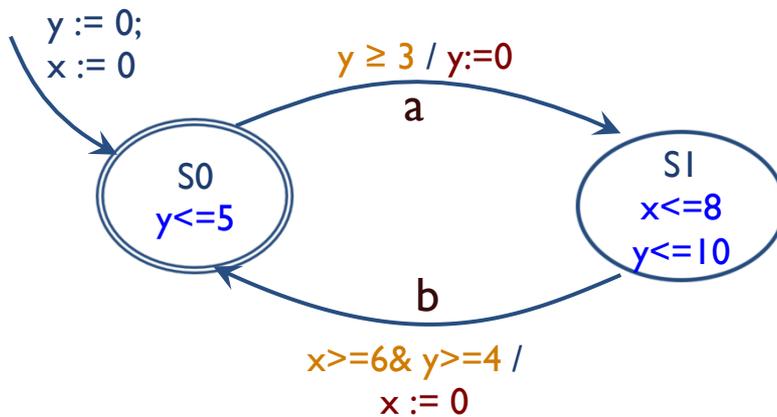
Transitions d'**action**: pour  $a \in \Sigma$ , on note  $(s, v) \xrightarrow{a} (s', v')$  s'il existe  $\langle s, a, \varphi, \lambda, s' \rangle \in T$  tel que:  $v$  satisfait  $\varphi$ ;  $v' = v[\lambda := 0]$ ;  $v'$  satisfait  $I(s')$ .



**XOR**



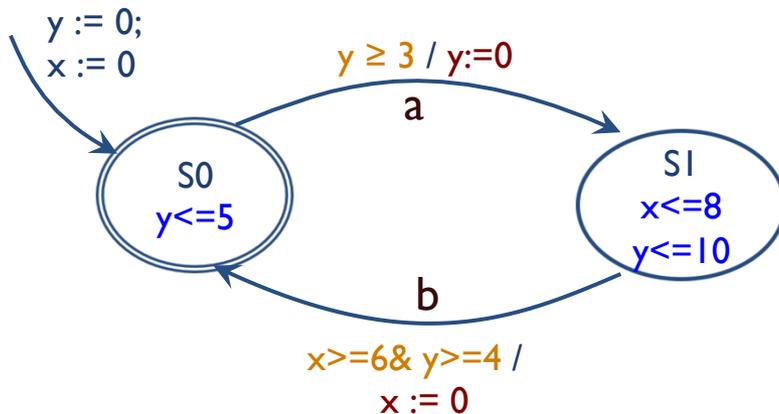
# Exemple



Quelques trajectoires:

$(s0, 0, 0) \xrightarrow{3} (s0, 3, 3) \xrightarrow{a} (s1, 3, 0) \xrightarrow{4}$   
 $(s1, 7, 4) \xrightarrow{b} (s0, 0, 4) \xrightarrow{l} (s0, 1, 5) \xrightarrow{a}$   
 $(s1, 1, 0) \dots$

# Exemple



Quelques trajectoires:

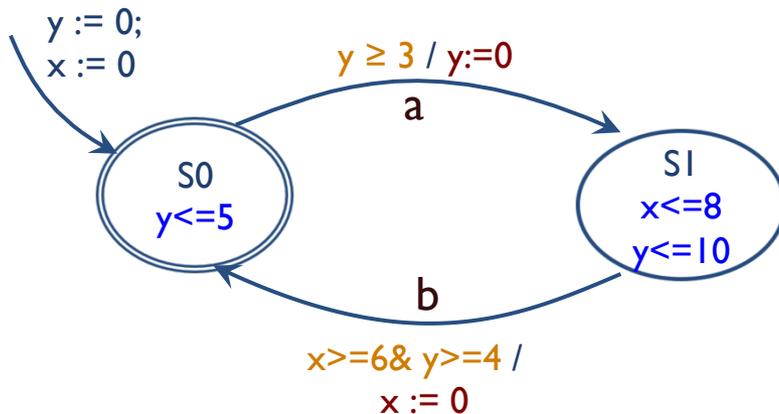
$(s0, 0, 0) \xrightarrow{3} (s0, 3, 3) \xrightarrow{a} (s1, 3, 0) \xrightarrow{4}$   
 $(s1, 7, 4) \xrightarrow{b} (s0, 0, 4) \xrightarrow{l} (s0, 1, 5) \xrightarrow{a}$   
 $(s1, 1, 0) \dots$

Correct ?

1.  $(s0, 0, 0) \xrightarrow{3.2} (s0, 3.2, 3.2) \xrightarrow{a} (s1, 3.2, 0)$   
 $\xrightarrow{4.1} (s1, 7.3, 4.1) \xrightarrow{b} (s0, 0, 4.1) \xrightarrow{0.1}$   
 $(s0, 0.1, 4.2) \xrightarrow{a} (s1, 0.1, 0) \dots$

2.  $(s0, 0, 0) \xrightarrow{3.5} (s0, 3.5, 3.5) \xrightarrow{a} (s1, 3.5, 0)$   
 $\xrightarrow{4.2} (s1, 7.7, 4.2) \xrightarrow{b} (s0, 0, 4.2) \xrightarrow{2}$   
 $(s0, 2, 6.2) \xrightarrow{a} (s1, 2, 0) \dots$

# Exemple



Quelques trajectoires:

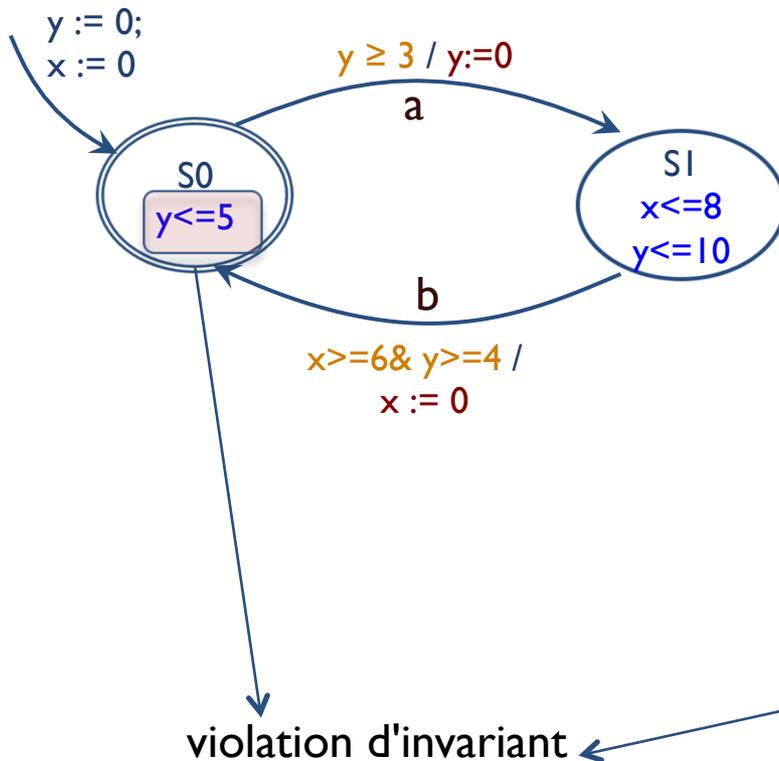
$(s0, 0, 0) \xrightarrow{3} (s0, 3, 3) \xrightarrow{a} (s1, 3, 0) \xrightarrow{4}$   
 $(s1, 7, 4) \xrightarrow{b} (s0, 0, 4) \xrightarrow{l} (s0, 1, 5) \xrightarrow{a}$   
 $(s1, 1, 0) \dots$

Correct ?

1.  $(s0, 0, 0) \xrightarrow{3.2} (s0, 3.2, 3.2) \xrightarrow{a} (s1, 3.2, 0)$   
 $\xrightarrow{4.1} (s1, 7.3, 4.1) \xrightarrow{b} (s0, 0, 4.1) \xrightarrow{0.1}$   
 $(s0, 0.1, 4.2) \xrightarrow{a} (s1, 0.1, 0) \dots$

2.  $(s0, 0, 0) \xrightarrow{3.5} (s0, 3.5, 3.5) \xrightarrow{a} (s1, 3.5, 0)$   
 $\xrightarrow{4.2} (s1, 7.7, 4.2) \xrightarrow{b} (s0, 0, 4.2) \xrightarrow{2}$   
 ~~$(s0, 2, 6.2) \xrightarrow{a} (s1, 2, 0) \dots$~~

# Exemple



Quelques trajectoires:

$(s0, 0, 0) \xrightarrow{3} (s0, 3, 3) \xrightarrow{a} (s1, 3, 0) \xrightarrow{4}$   
 $(s1, 7, 4) \xrightarrow{b} (s0, 0, 4) \xrightarrow{l} (s0, 1, 5) \xrightarrow{a}$   
 $(s1, 1, 0) \dots$

Correct ?

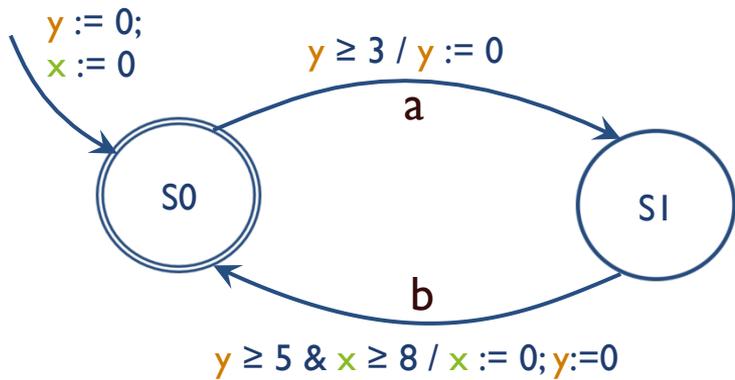
1.  $(s0, 0, 0) \xrightarrow{3.2} (s0, 3.2, 3.2) \xrightarrow{a} (s1, 3.2, 0)$   
 $\xrightarrow{4.1} (s1, 7.3, 4.1) \xrightarrow{b} (s0, 0, 4.1) \xrightarrow{0.1}$   
 $(s0, 0.1, 4.2) \xrightarrow{a} (s1, 0.1, 0) \dots$

2.  $(s0, 0, 0) \xrightarrow{3.5} (s0, 3.5, 3.5) \xrightarrow{a} (s1, 3.5, 0)$   
 $\xrightarrow{4.2} (s1, 7.7, 4.2) \xrightarrow{b} (s0, 0, 4.2) \xrightarrow{2}$   
 ~~$(s0, 2, 6.2) \xrightarrow{a} (s1, 2, 0) \dots$~~

# Exercice :

## Trace d'un automate temporel

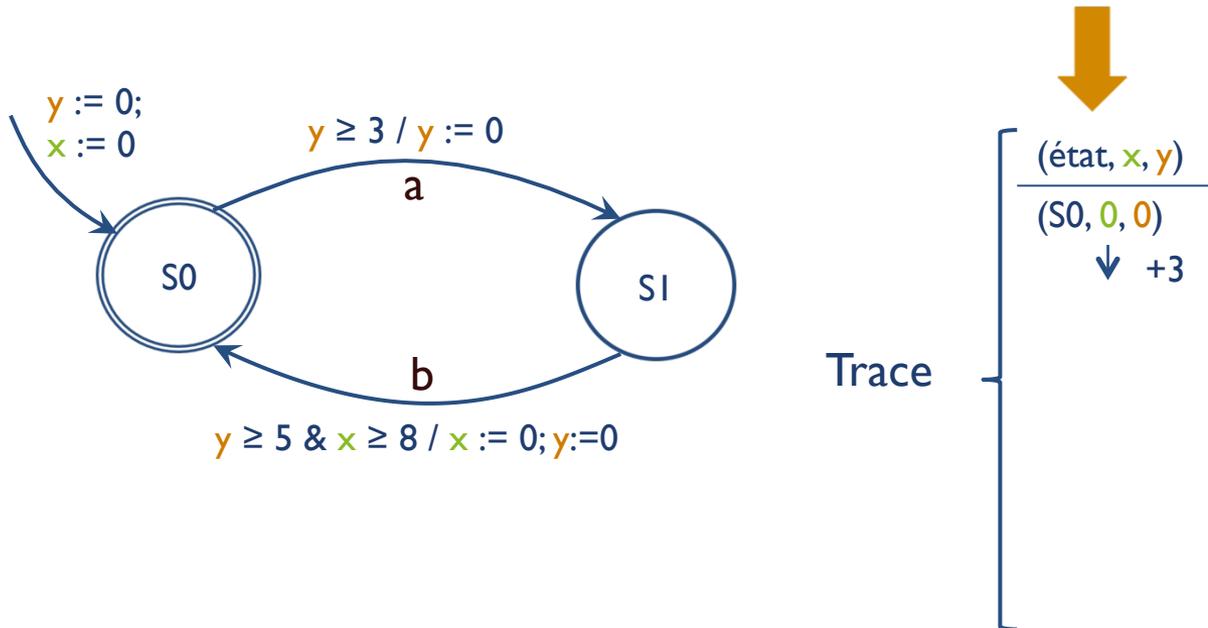
---



# Exercice :

## Trace d'un automate temporel

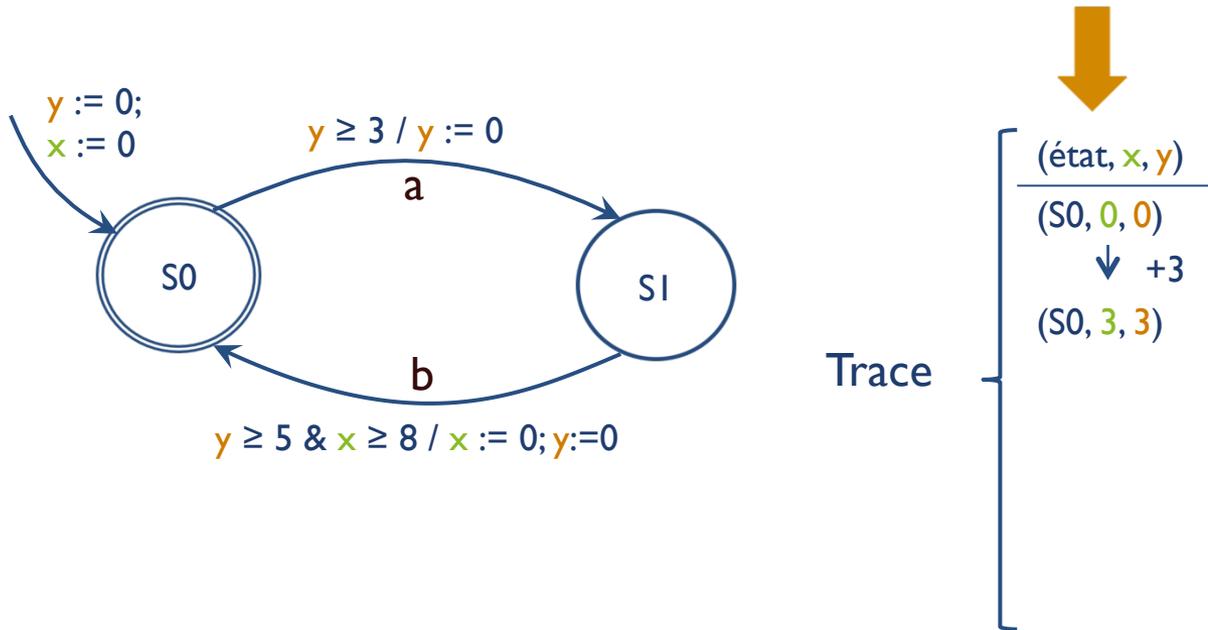
---



# Exercice :

## Trace d'un automate temporisé

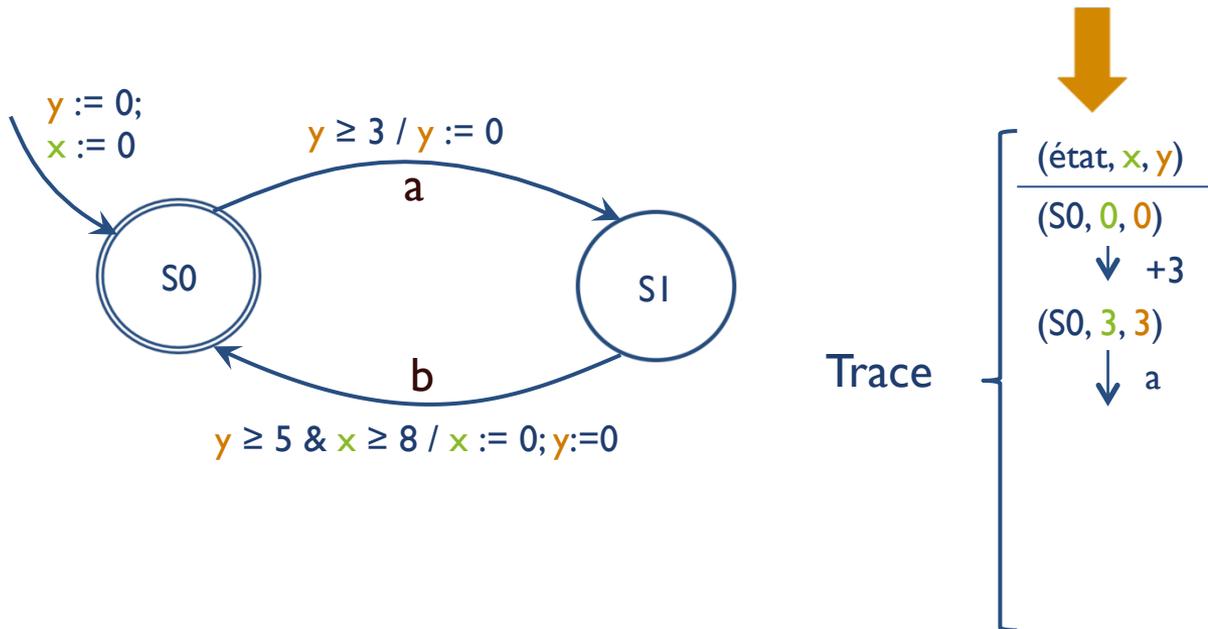
---



# Exercice :

## Trace d'un automate temporel

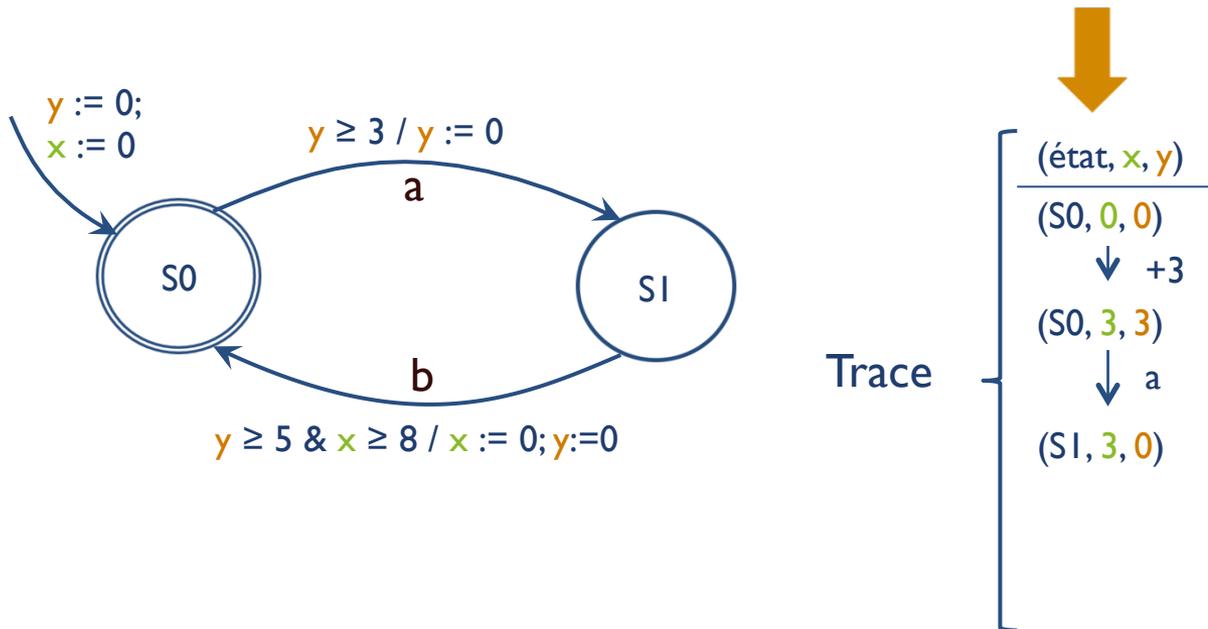
---



# Exercice :

## Trace d'un automate temporel

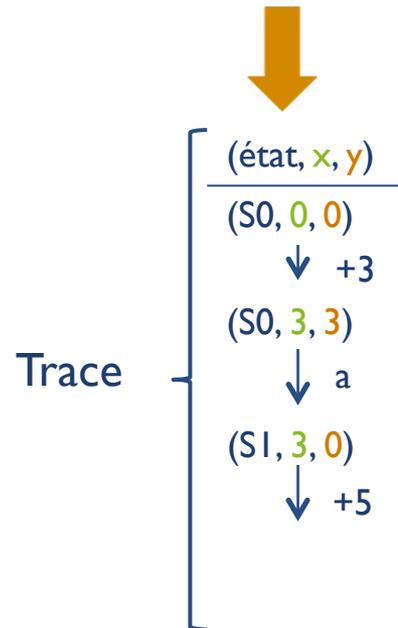
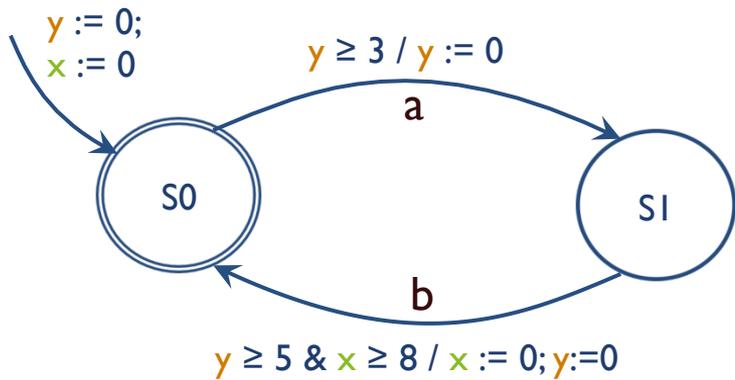
---



# Exercice :

## Trace d'un automate temporel

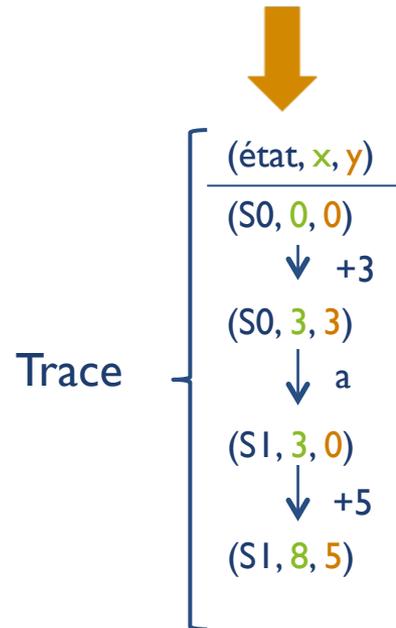
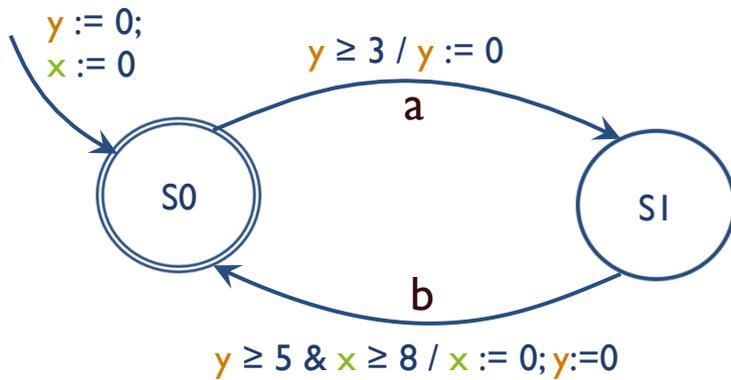
---



# Exercice :

## Trace d'un automate temporel

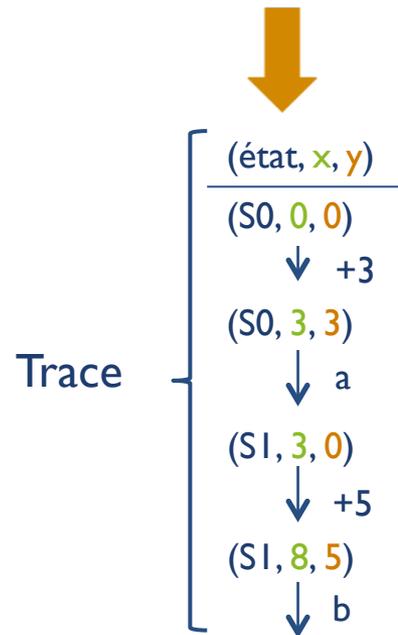
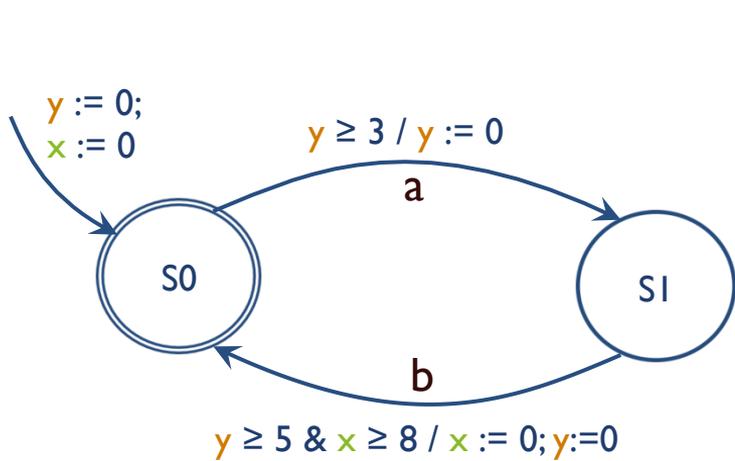
---



# Exercice :

## Trace d'un automate temporel

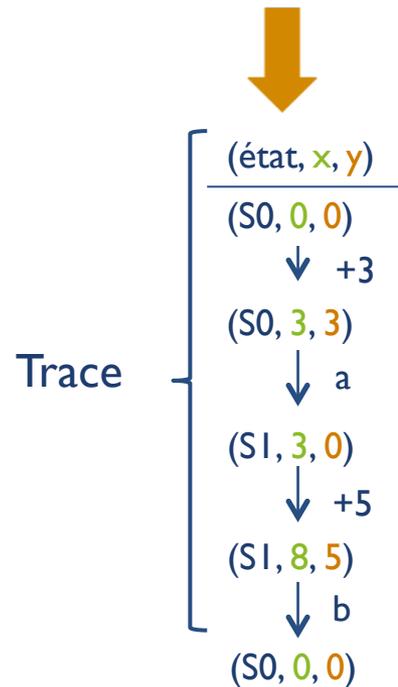
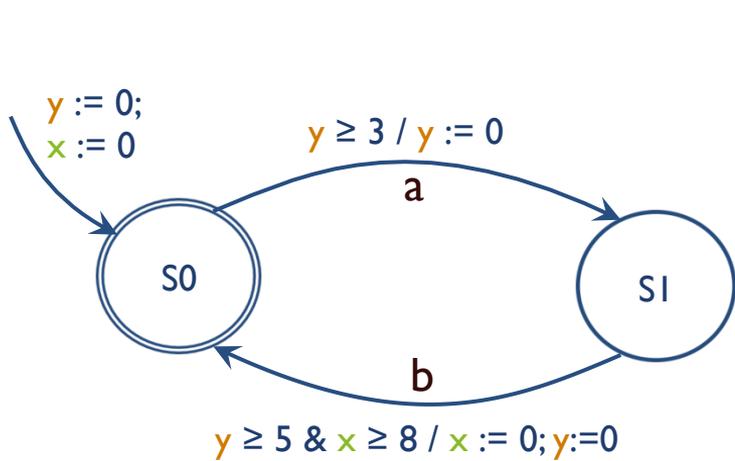
---



# Exercice :

## Trace d'un automate temporel

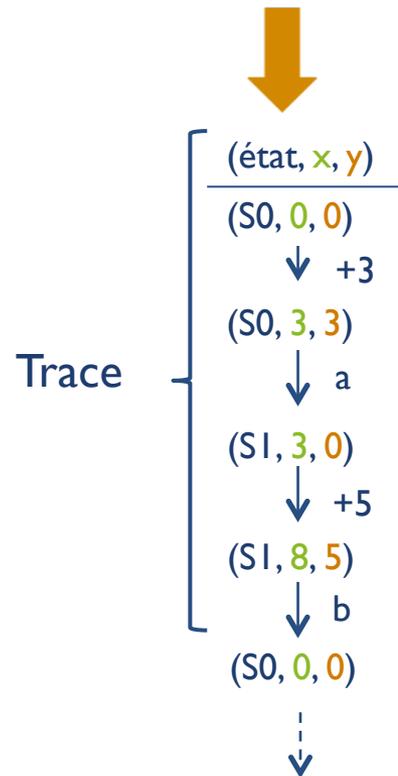
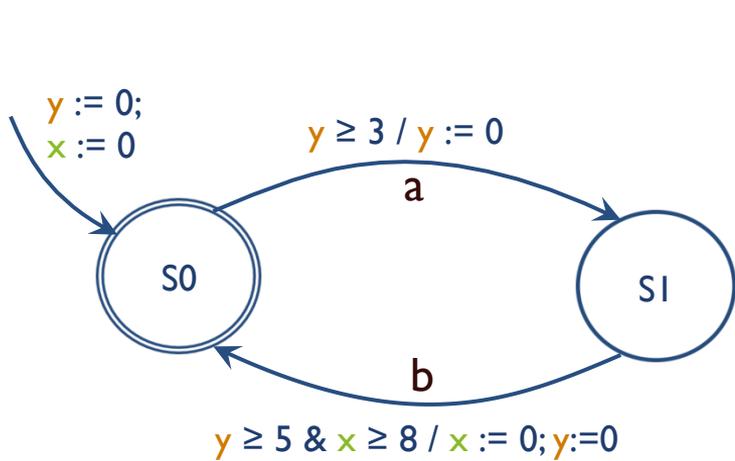
---



# Exercice :

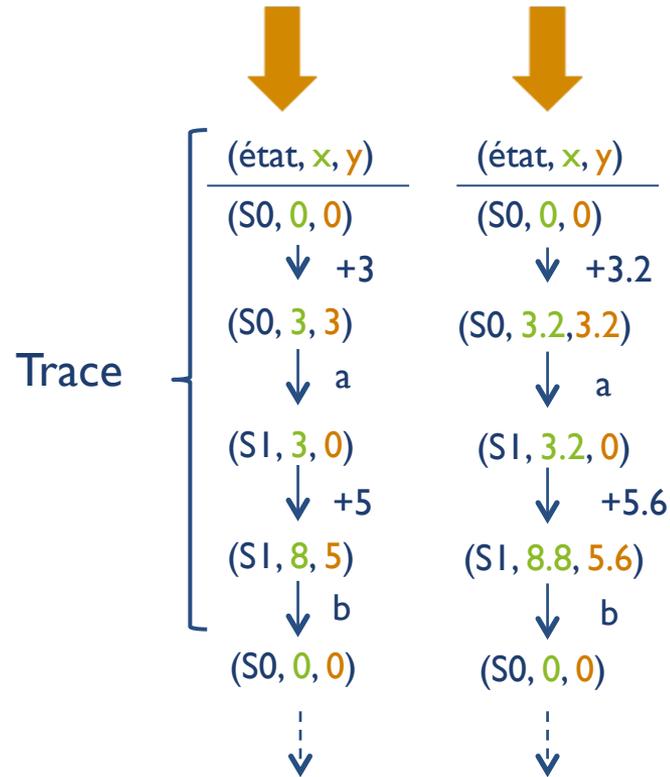
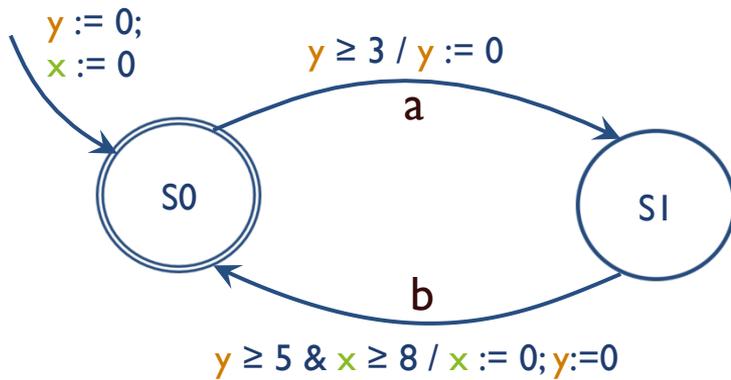
## Trace d'un automate temporel

---

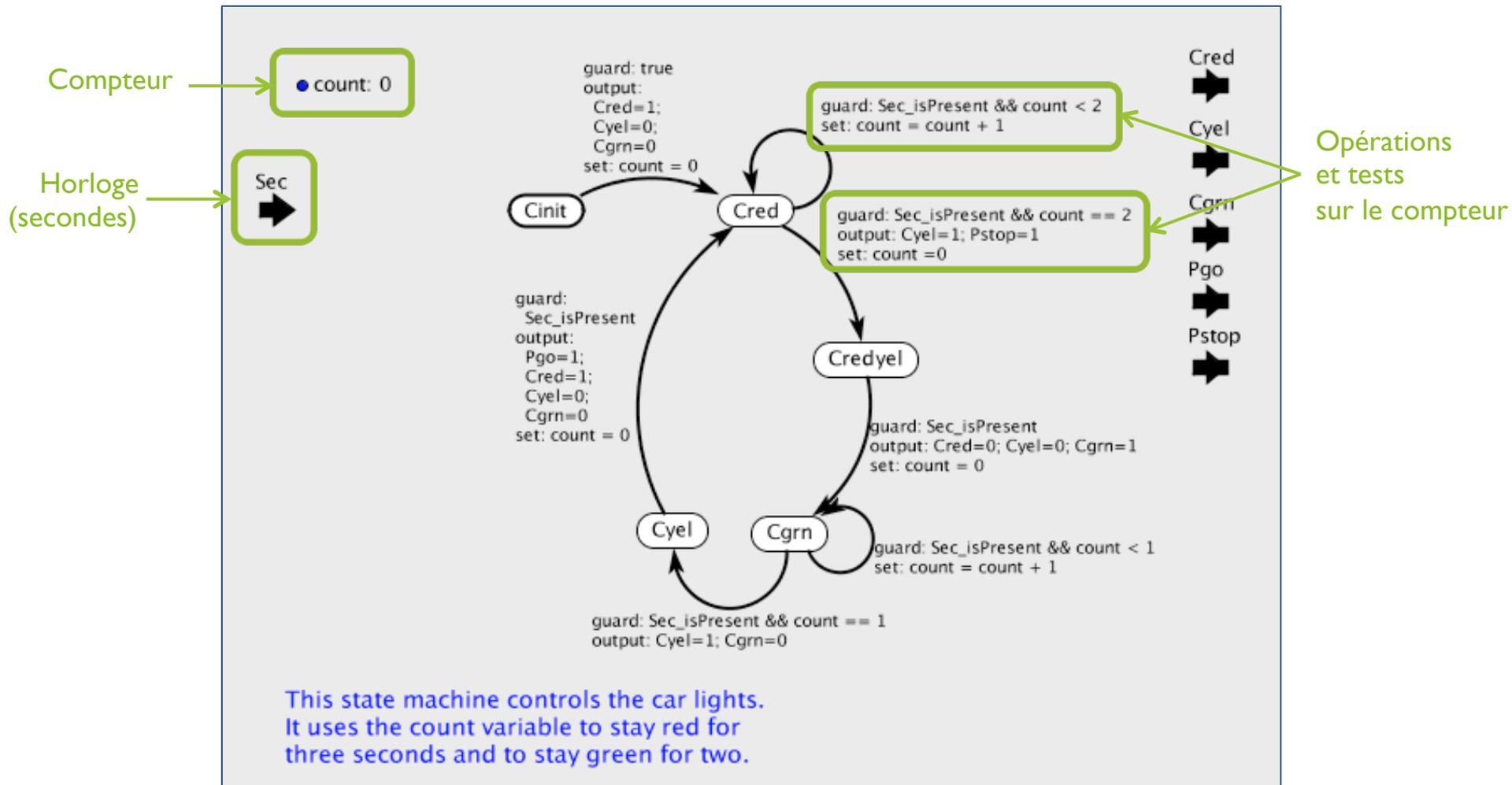


# Exercice :

## Trace d'un automate temporisé

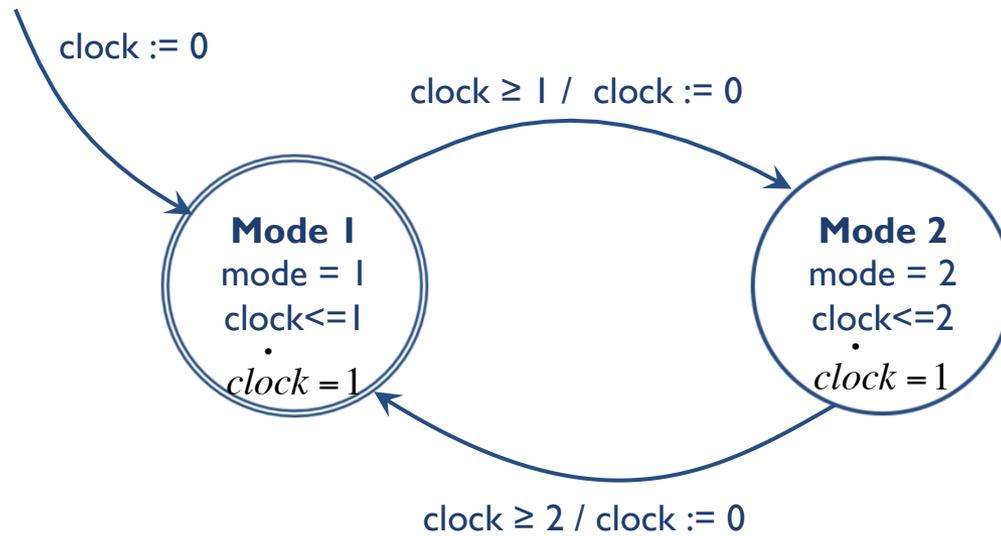


# Automate temporisé?



# Exemple : Alternateur

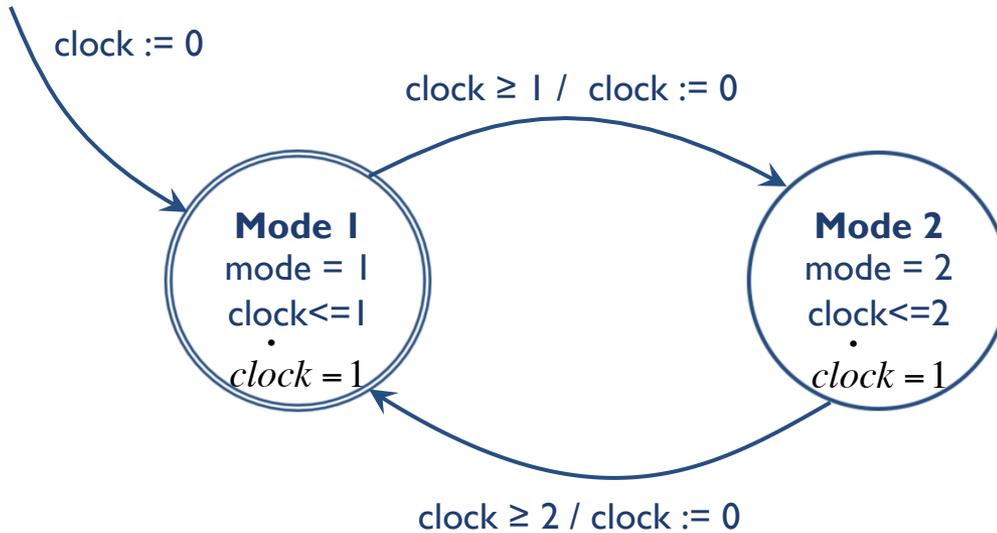
---



# Exercice :

## Trace de l'alternateur

---

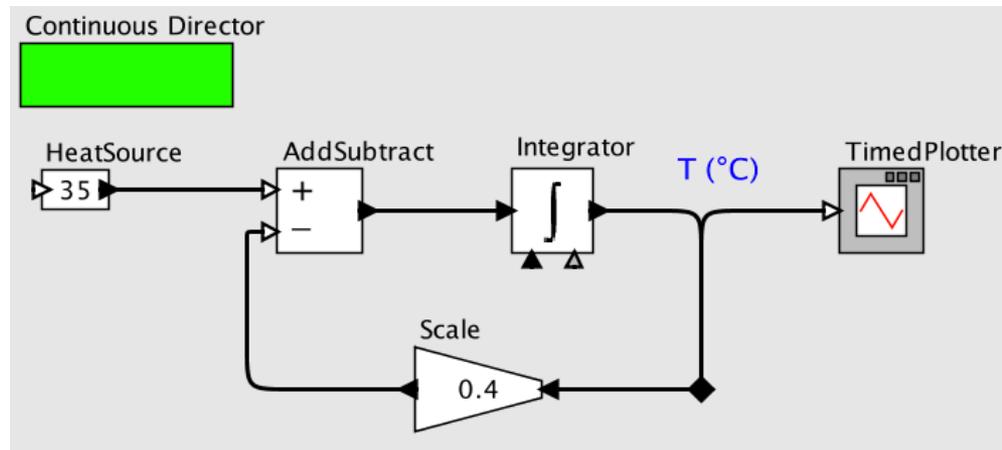


Montrer clock(t) et mode(t) sur un graphique

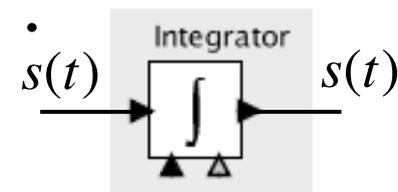


# Lire des équations différentielles

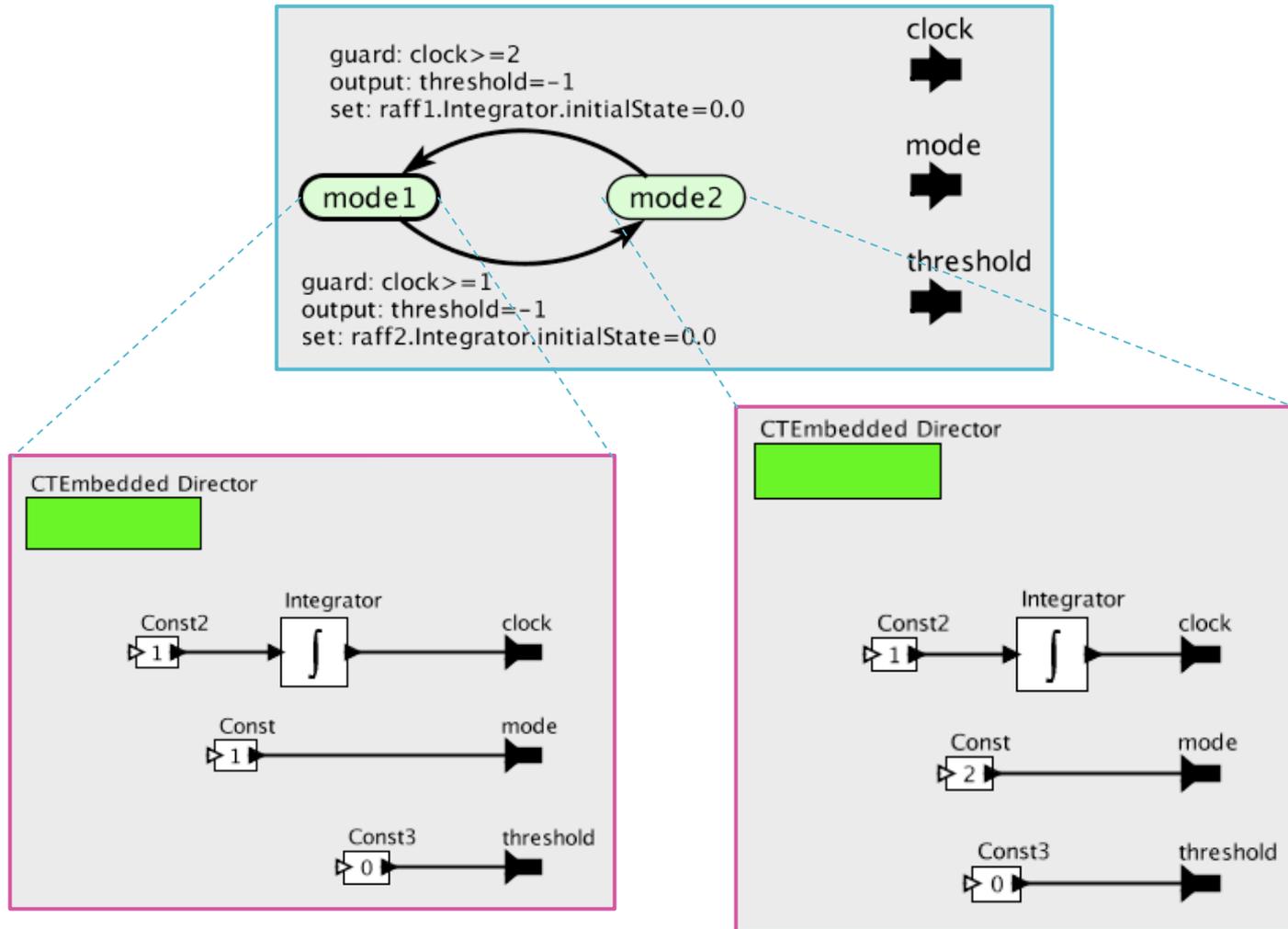
- ▶ Objectif = lire un modèle d'équation différentielle au format schéma-bloc



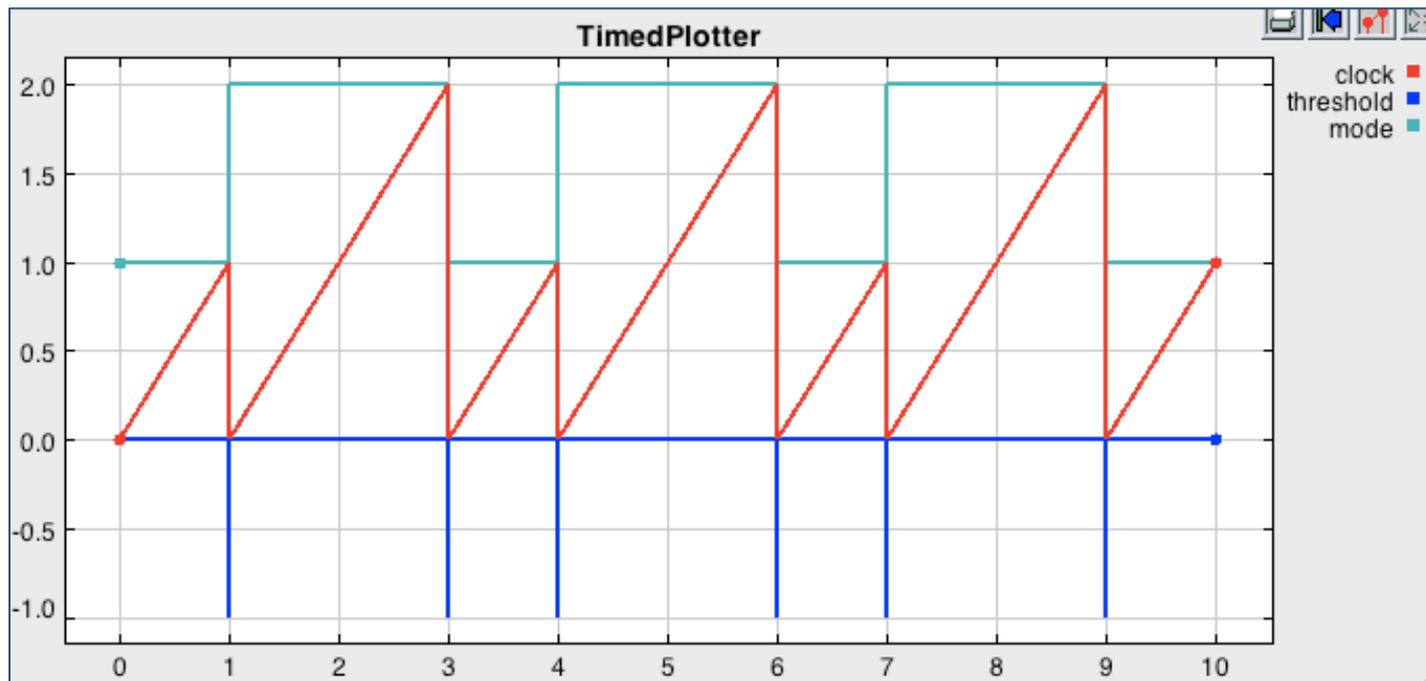
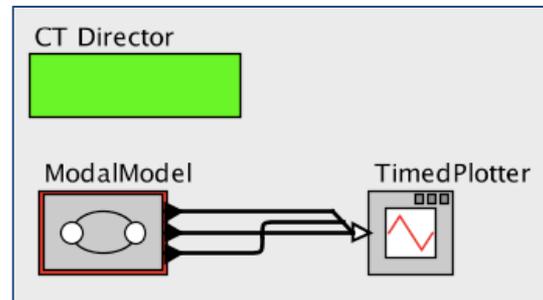
- ▶ Éléments de modélisation utilisés :
  - ▶ **Constante** : émet un signal constant continu
  - ▶ **Intégrateur** : produit en sortie le signal intégrale du signal d'entrée
    - ▶ Prend en paramètre la valeur initiale du signal de sortie ( $s_0$ )
  - ▶ **Relation** : égalité entre deux signaux



# Exemple : Alternateur, modèle modal



# Exemple : Alternateur, simulation



# Modélisation: Composition de AT

---

- Soient  $A_1 = (\Sigma_1, S_1, s_{01}, X_1, I_1, T_1)$  et  $A_2 = (\Sigma_2, S_2, s_{02}, X_2, I_2, T_2)$  avec  $X_1 \cap X_2 = \emptyset$ , alors  $A_1 \parallel A_2$  est l'automate temporisé

$$(\Sigma_1 \cup \Sigma_2, S_1 \times S_2, (s_{01}, s_{02}), X_1 \cup X_2, I, T) \text{ avec}$$

- $I(s_1, s_2) = I_1(s_1) \wedge I_2(s_2)$

# Modélisation: Composition de AT

---

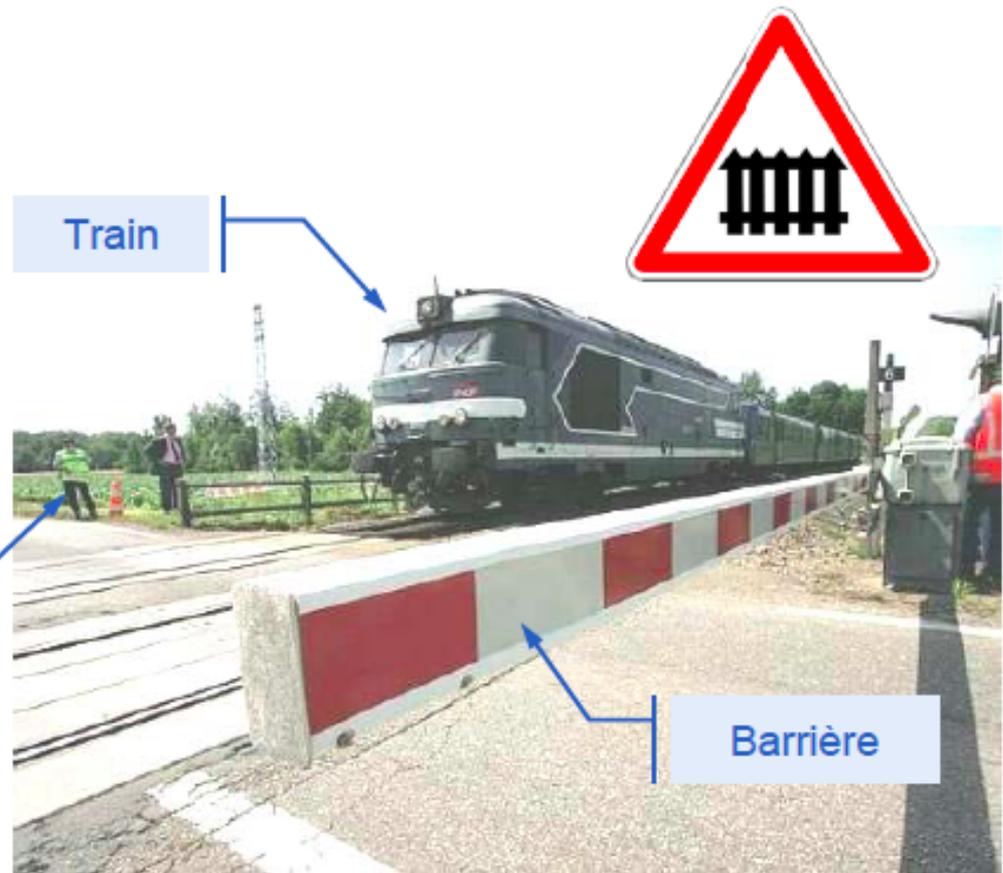
- Soient  $A_1 = (\Sigma_1, S_1, s_{01}, X_1, I_1, T_1)$  et  $A_2 = (\Sigma_2, S_2, s_{02}, X_2, I_2, T_2)$  avec  $X_1 \cap X_2 = \emptyset$ , alors  $A_1 \parallel A_2$  est l'automate temporisé

$$(\Sigma_1 \cup \Sigma_2, S_1 \times S_2, (s_{01}, s_{02}), X_1 \cup X_2, I, T) \text{ avec}$$

- $I(s_1, s_2) = I_1(s_1) \wedge I_2(s_2)$
- T est défini par
  1. Pour  $a \in \Sigma_1 \cap \Sigma_2$   
 $\langle s_1, a, \Phi_1, \lambda_1, s_1' \rangle \in T_1$  et  $\langle s_2, a, \Phi_2, \lambda_2, s_2' \rangle \in T_2 \rightarrow$   
 $\langle (s_1, s_2), a, \Phi_1 \wedge \Phi_2, \lambda_1 \cup \lambda_2, (s_1', s_2') \rangle \in T$
  2. Pour  $a \in \Sigma_1 - \Sigma_2$   
 $\langle s, a, \Phi, \lambda, s' \rangle \in T_1$  et  $t \in S_2 \rightarrow \langle (s, t), a, \Phi, \lambda, (s', t) \rangle \in T$
  3. symétriquement

# Exemple (Train, Barrière, Contrôleur)

- Comportement temporel d'un système où 3 entités coopèrent



Train

Contrôleur  
(garde barrière)

Barrière

# Exemple (Train, Barrière, Contrôleur)

## Description du problème

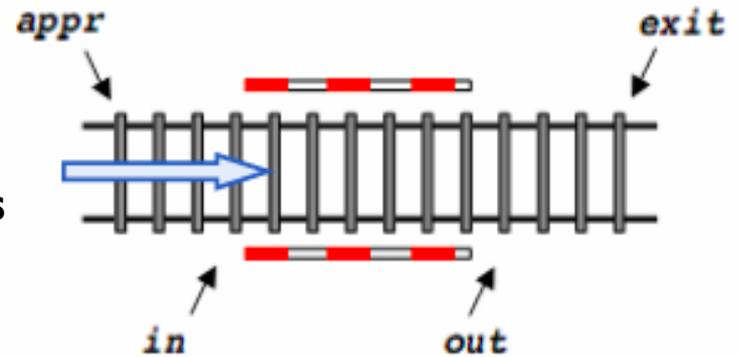
- ◆ **Systeme faisant coopérer 3 composants**

- ◆ Le train

- ◆ La barrière

- ◆ Le contrôleur

- ◆ **Systeme=train || barrière || contrôleur**

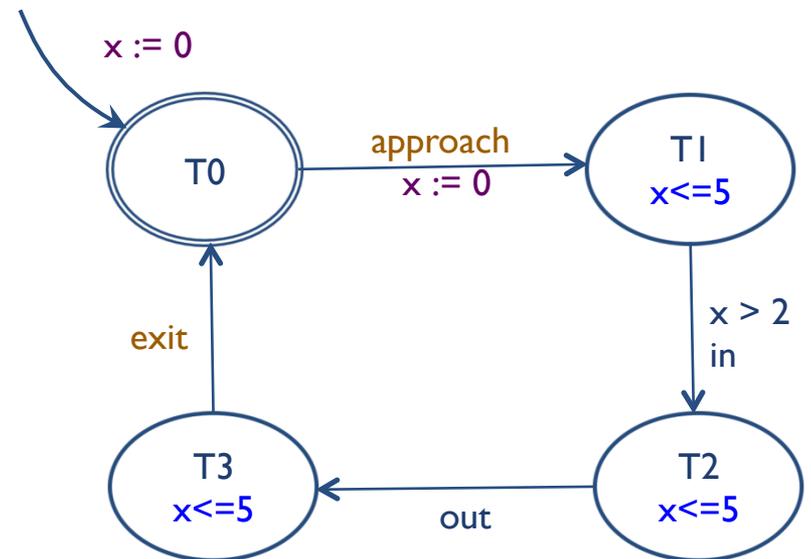
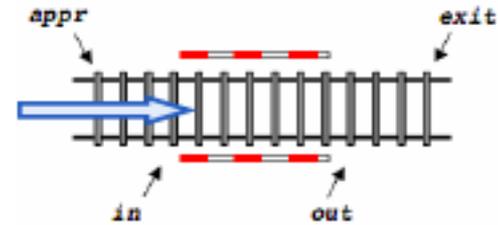


- ◆ **Propriété à vérifier: quand le train est dans la section du passage à niveau, la barrière est fermée.**

# Exemple (Train, Barrière, Contrôleur)

## Train

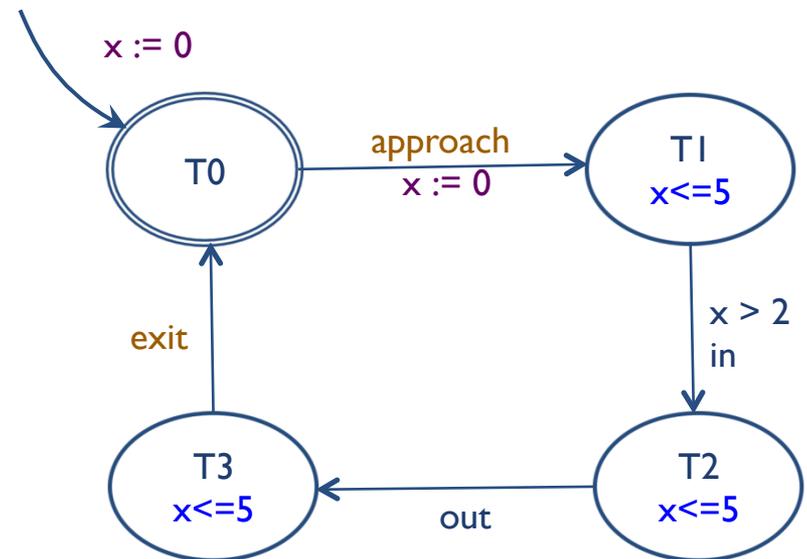
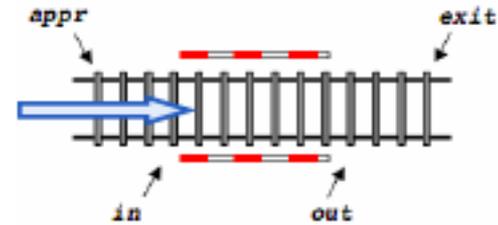
- Le train communique par deux signaux **approach** et **exit**.
- Le train envoie le signal **approach** au moins **? minutes** avant d'entrer dans le passage.
- Le temps maximum entre les événements **approach** et **out** est de **? minutes**.



# Exemple (Train, Barrière, Contrôleur)

## Train

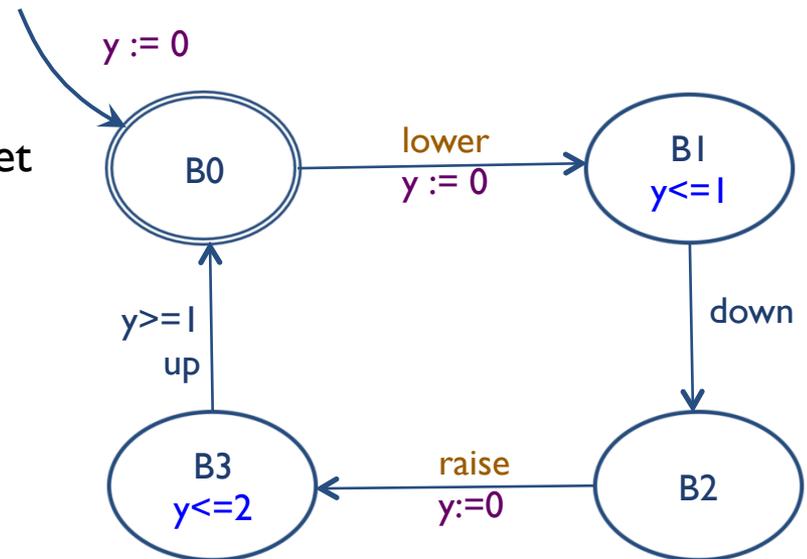
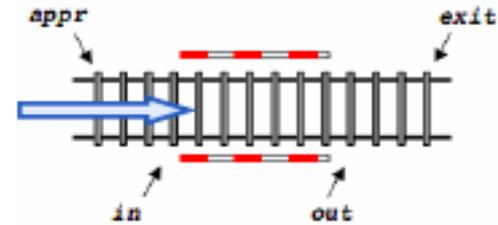
- Le train communique par deux signaux **approach** et **exit**.
- Le train envoie le signal **approach** au moins **2 minutes** avant d'entrer dans le passage.
- Le temps maximum entre les événements **approach** et **out** est de **5 minutes**.



# Exemple (Train, Barrière, Contrôleur)

## Barrière

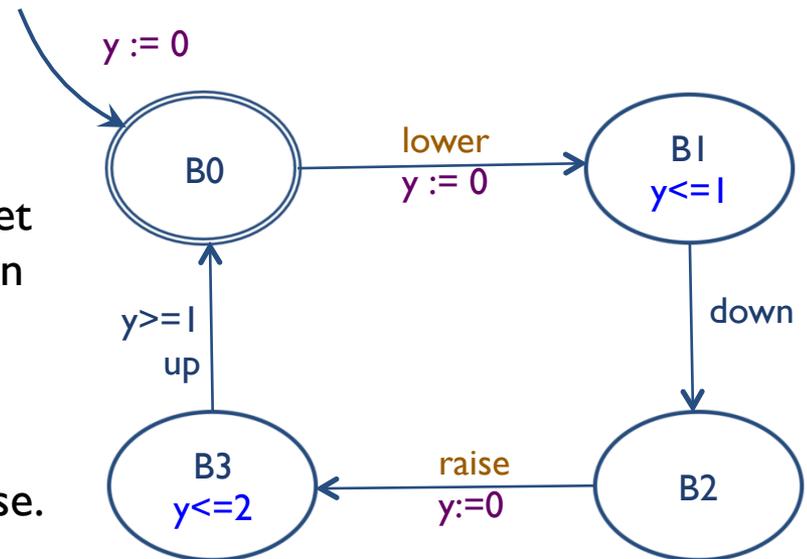
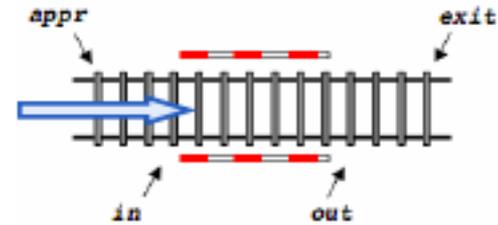
- Le barrière communique par deux signaux **lower** et **raise**.
- Quand le barrière reçoit le signal **lower**, elle atteint la position basse en ? minutes.
- Quand la barrière reçoit le signal **raise**, elle met ? minutes pour atteindre la position haute.
- Pendant l'absence de signal, la barrière peut rester ? minutes en position haute ou basse.



# Exemple (Train, Barrière, Contrôleur)

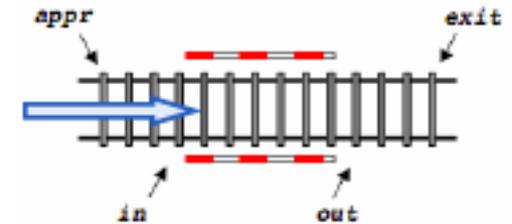
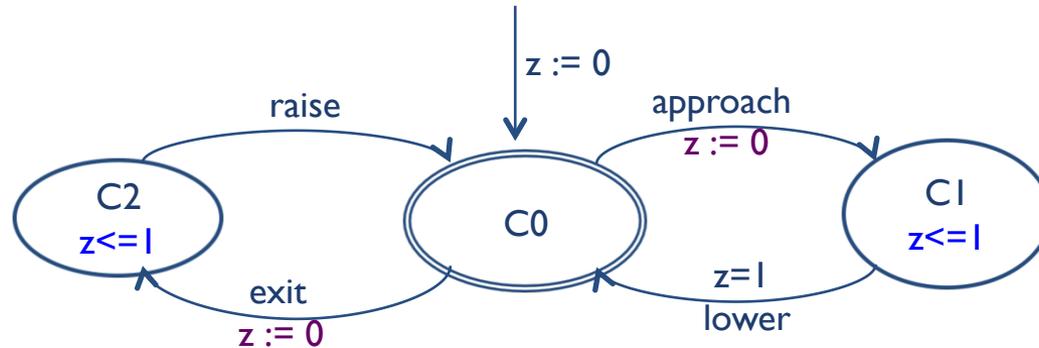
## Barrière

- Le barrière communique par deux signaux **lower** et **raise**.
- Quand le barrière reçoit le signal **lower**, elle atteint la position basse en **1 minute au maximum**.
- Quand la barrière reçoit le signal **raise**, elle met **entre 1 et 2 minutes** pour atteindre la position haute.
- Pendant l'absence de signal, la barrière peut rester **indéfiniment** en position haute ou basse.



# Exemple (Train, Barrière, Contrôleur)

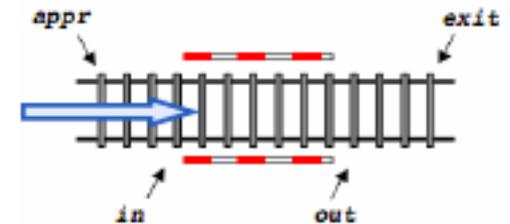
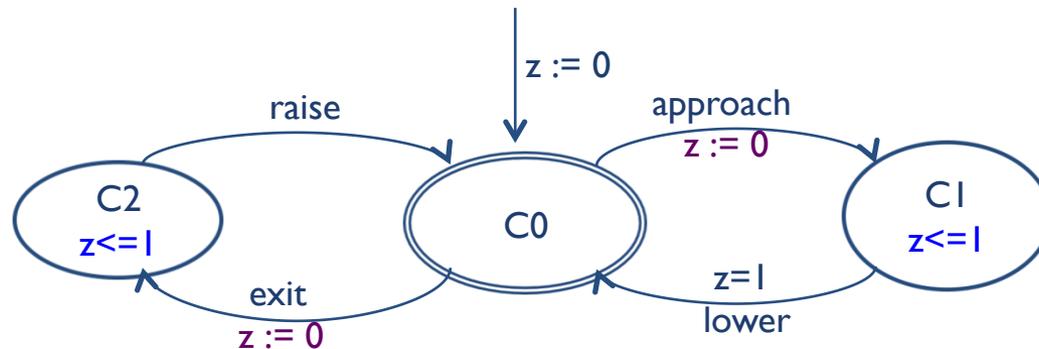
## Contrôleur



- Le contrôleur communique par **approach** et **exit** avec le train et par **lower** et **raise** avec la barrière.
- Lorsqu'il reçoit **approach** du train, le contrôleur répond, en ? minute, en envoyant **lower** à la barrière.
- Lorsqu'il reçoit **exit** du train, le contrôleur répond, en ? minute, en envoyant **raise** à la barrière.

# Exemple (Train, Barrière, Contrôleur)

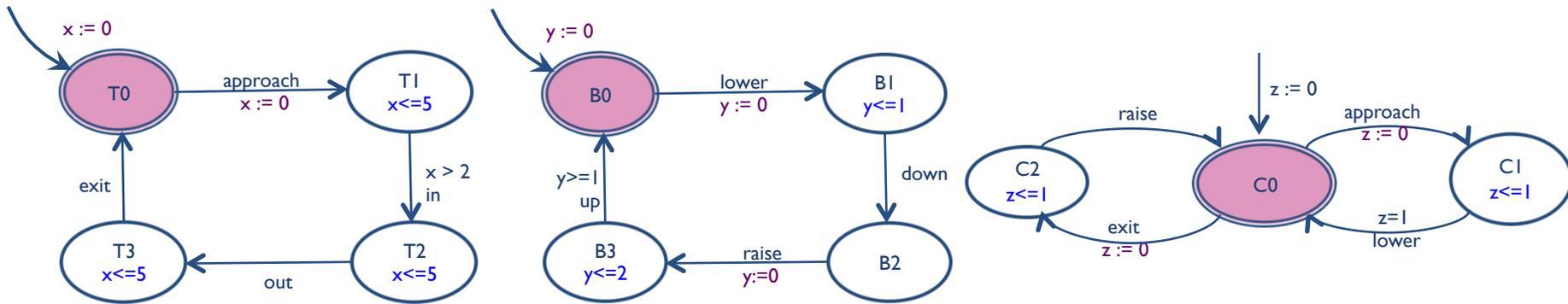
## Contrôleur



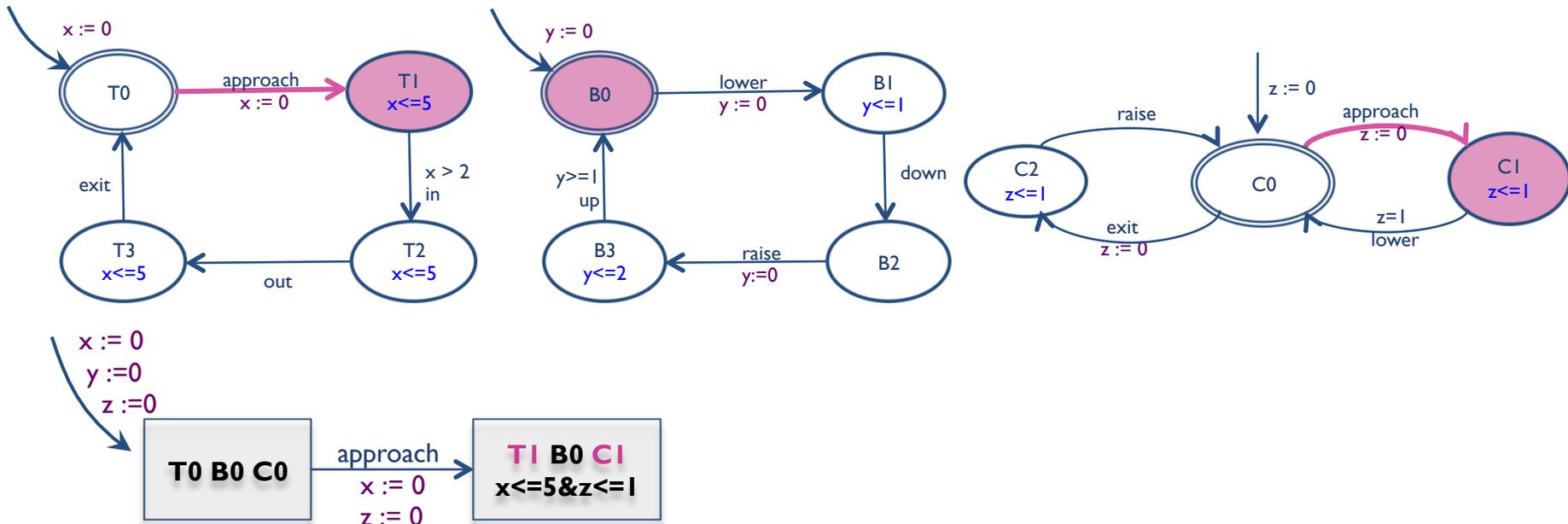
- Le contrôleur communique par **approach** et **exit** avec le train et par **lower** et **raise** avec la barrière.
- Lorsqu'il reçoit **approach** du train, le contrôleur répond, en **1 minute exactement**, en envoyant **lower** à la barrière.
- Lorsqu'il reçoit **exit** du train, le contrôleur répond, en **1 minute au maximum**, en envoyant **raise** à la barrière.

# Exemple (Train, Barrière, Contrôleur)

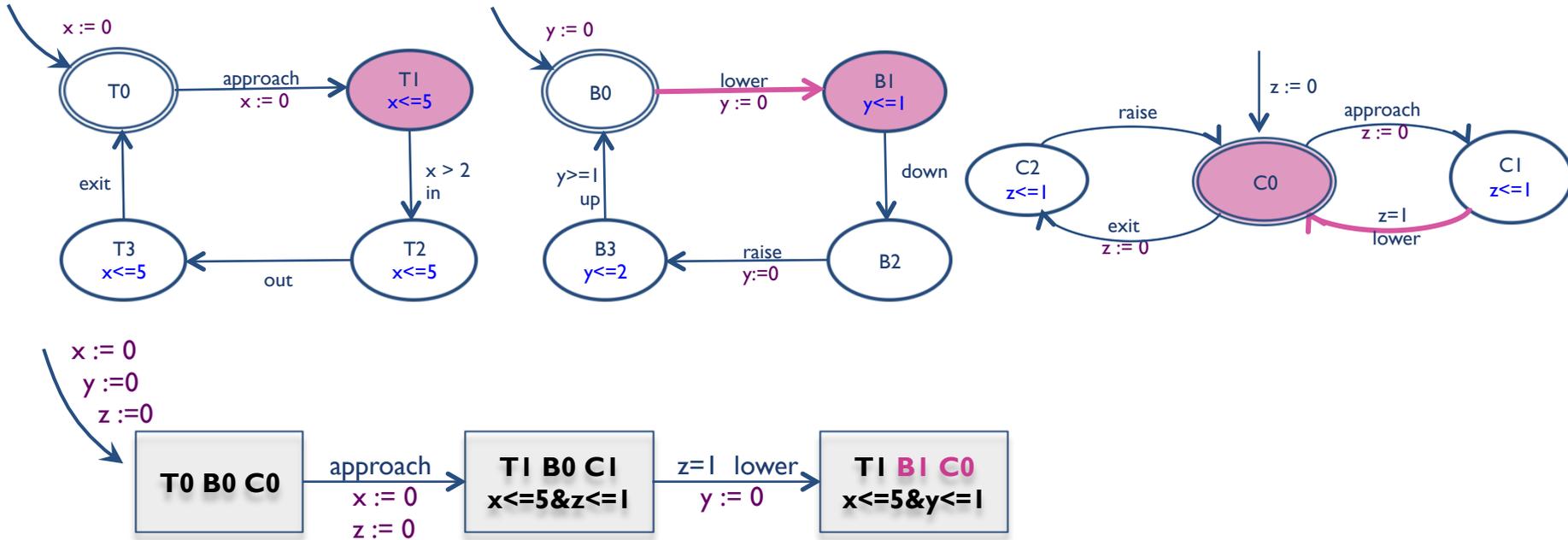
## Composition ?



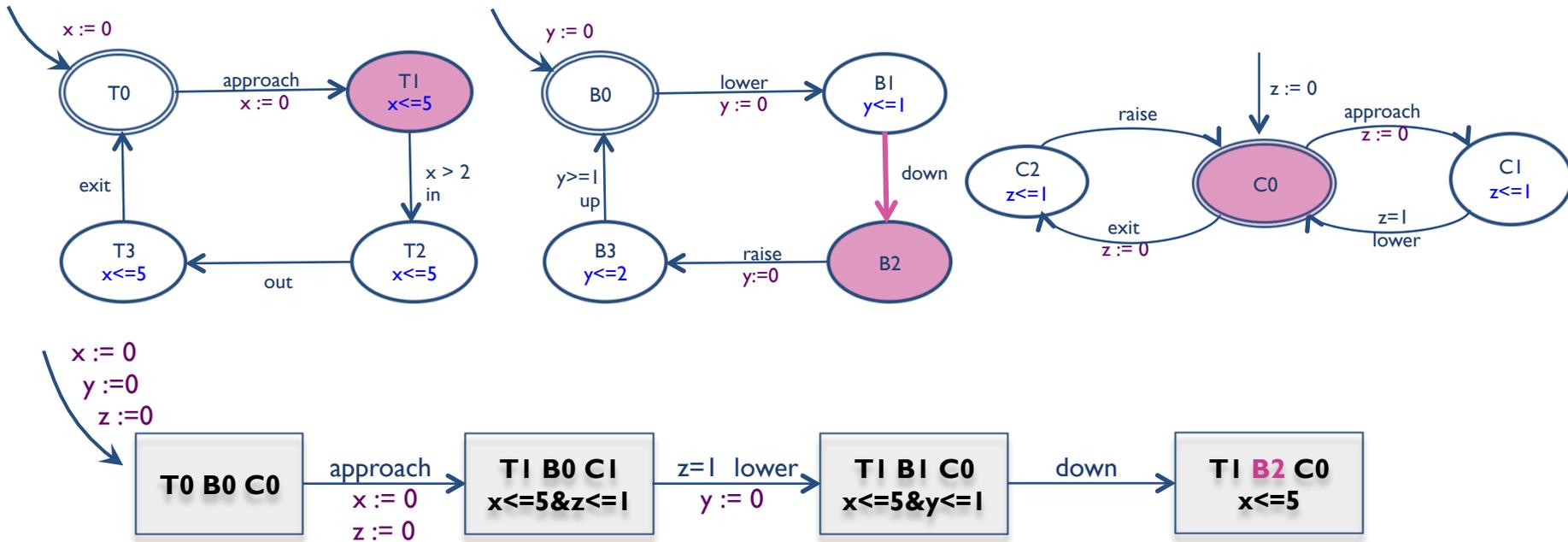
# Exemple (Train, Barrière, Contrôleur)



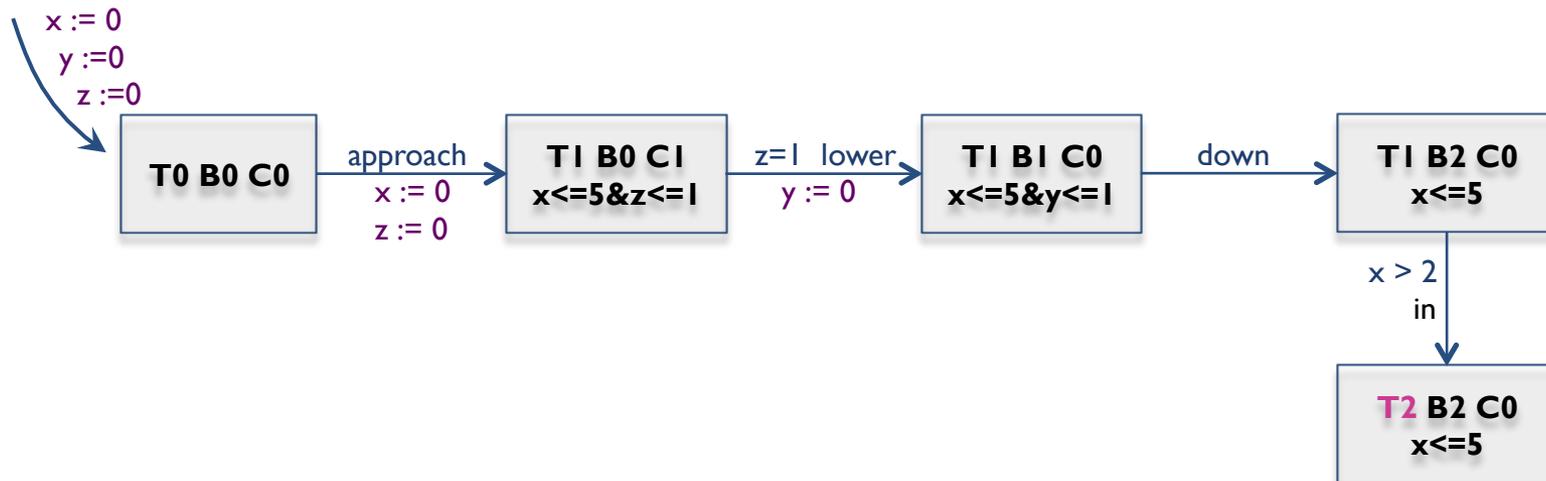
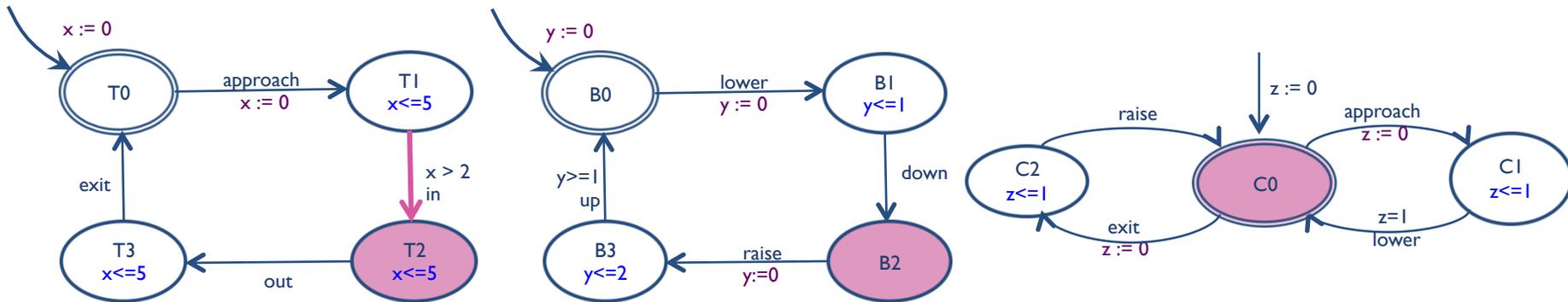
# Exemple (Train, Barrière, Contrôleur)



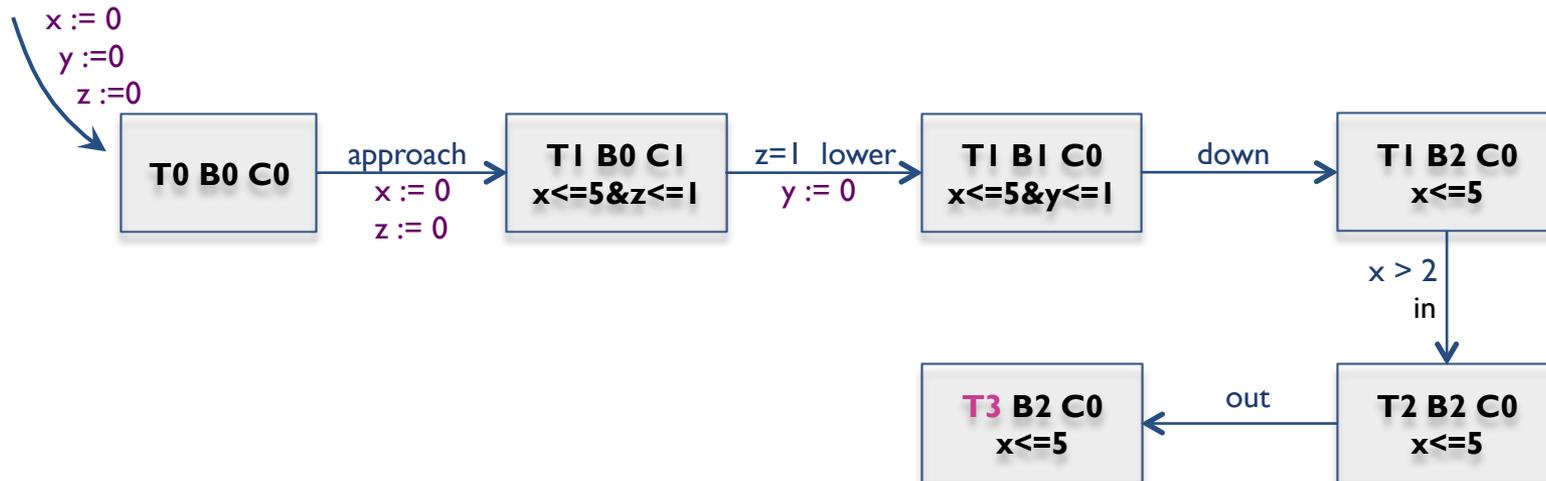
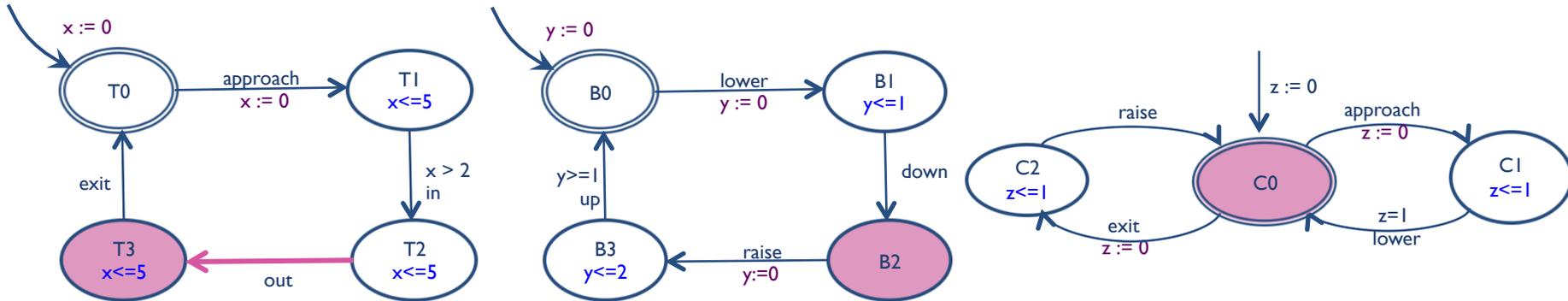
# Exemple (Train, Barrière, Contrôleur)



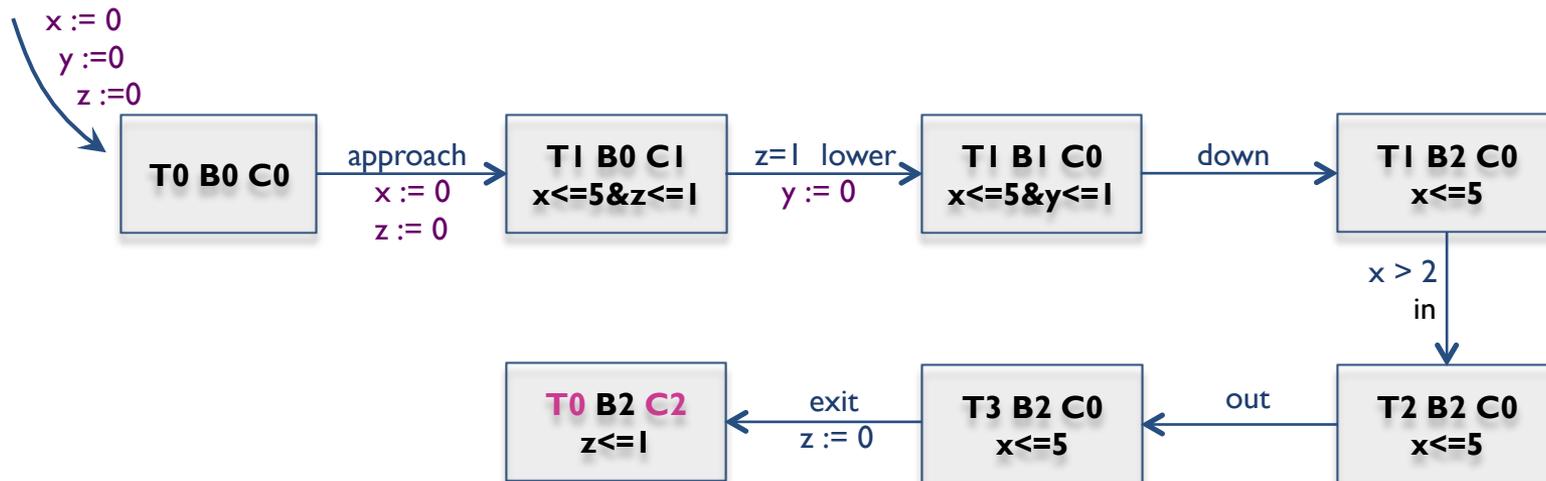
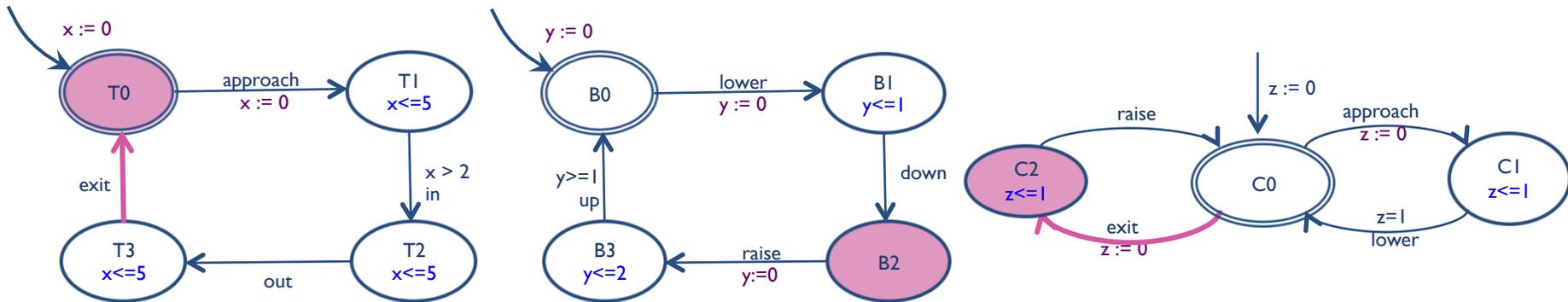
# Exemple (Train, Barrière, Contrôleur)



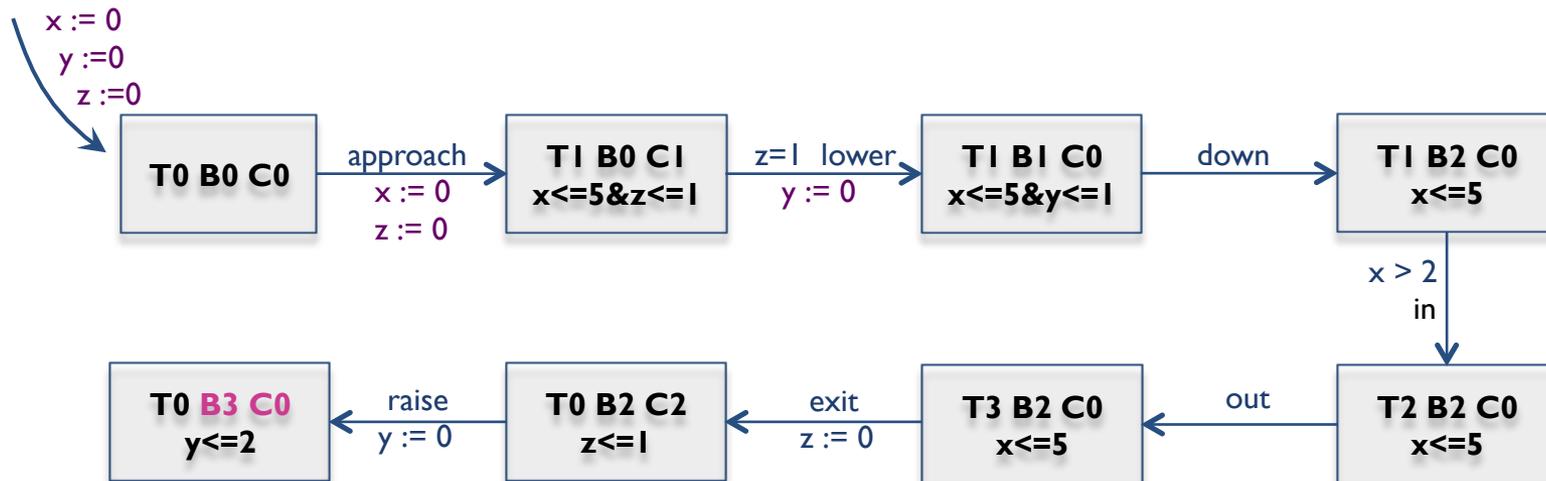
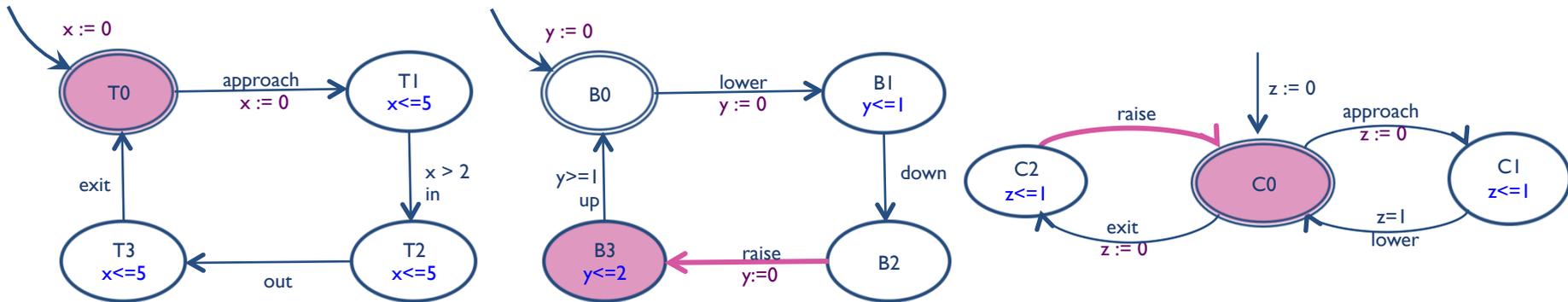
# Exemple (Train, Barrière, Contrôleur)



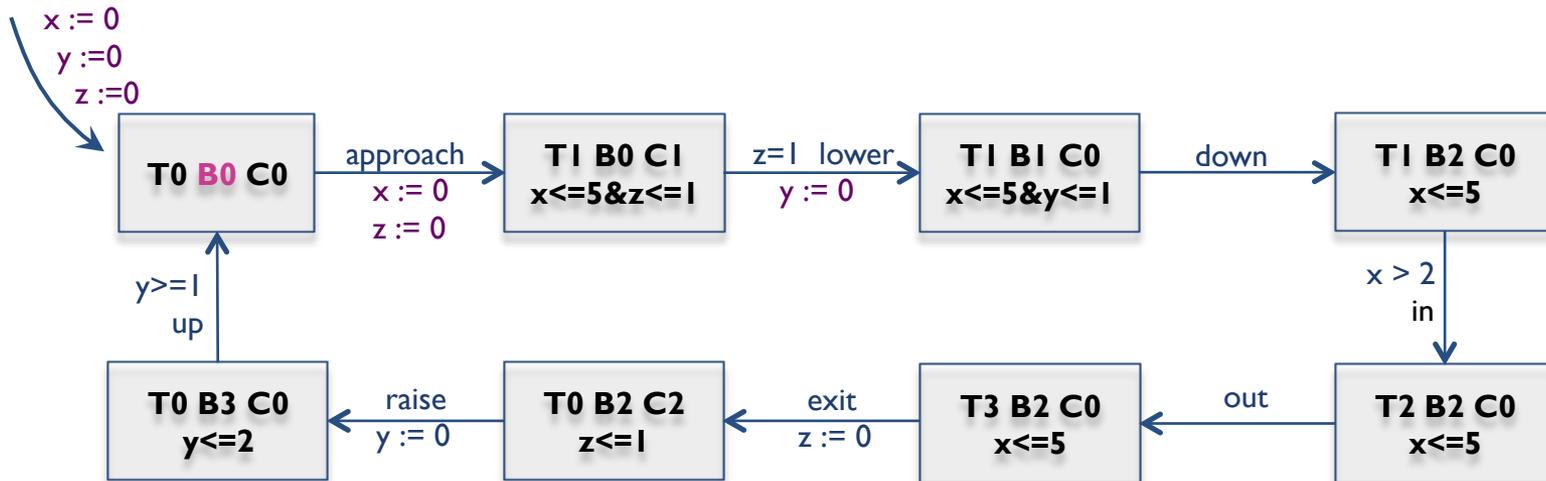
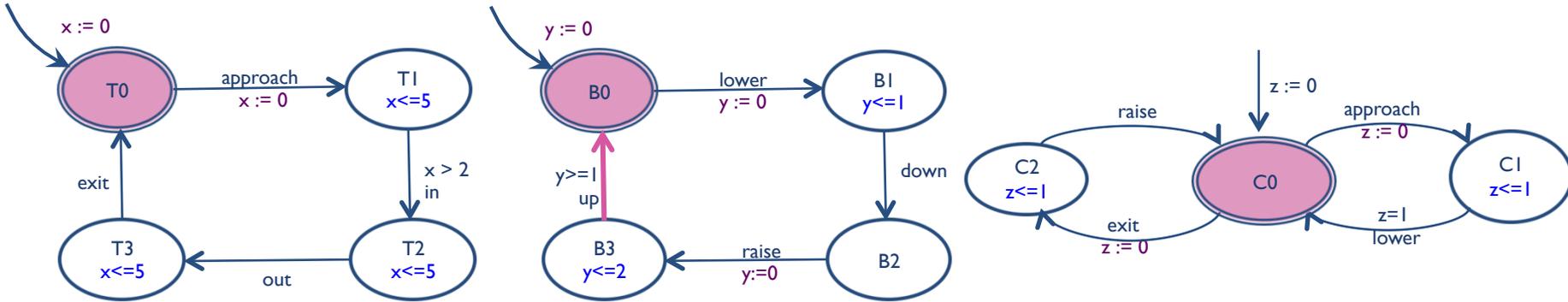
# Exemple (Train, Barrière, Contrôleur)



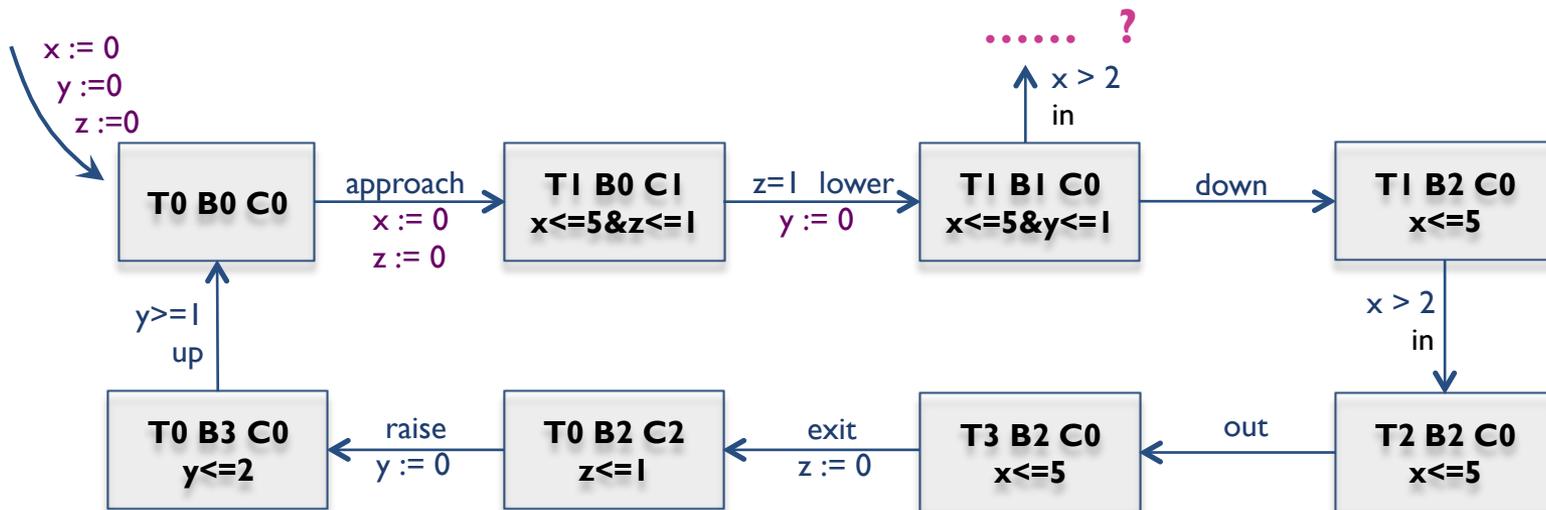
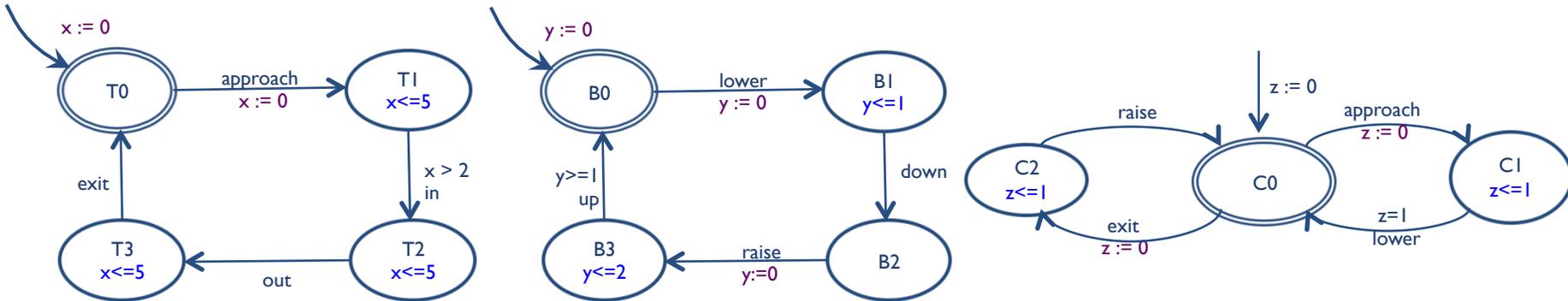
# Exemple (Train, Barrière, Contrôleur)



# Exemple (Train, Barrière, Contrôleur)

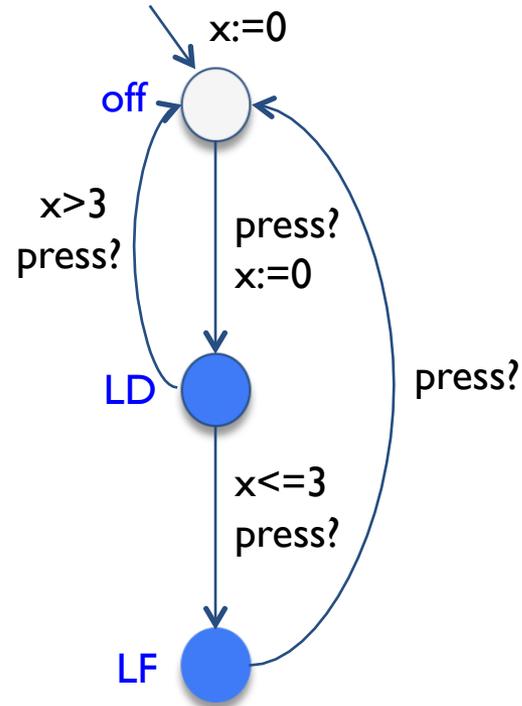


# Exemple (Train, Barrière, Contrôleur)

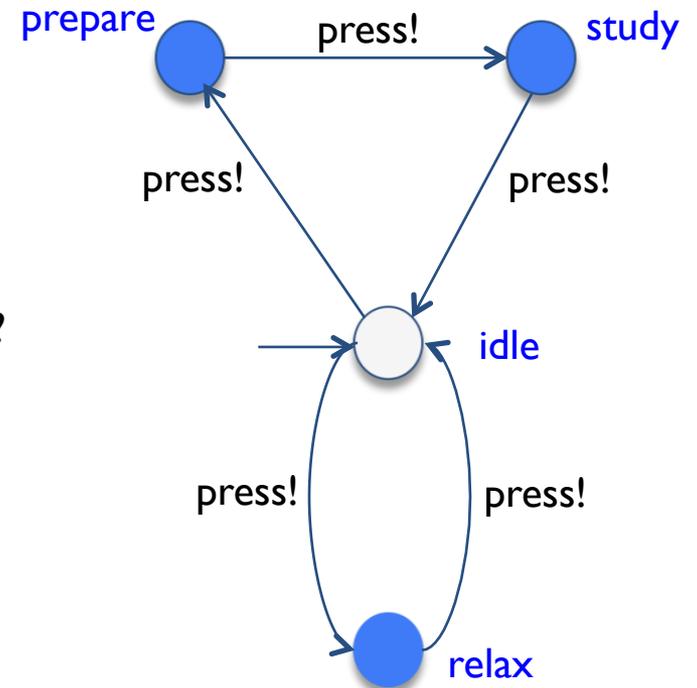
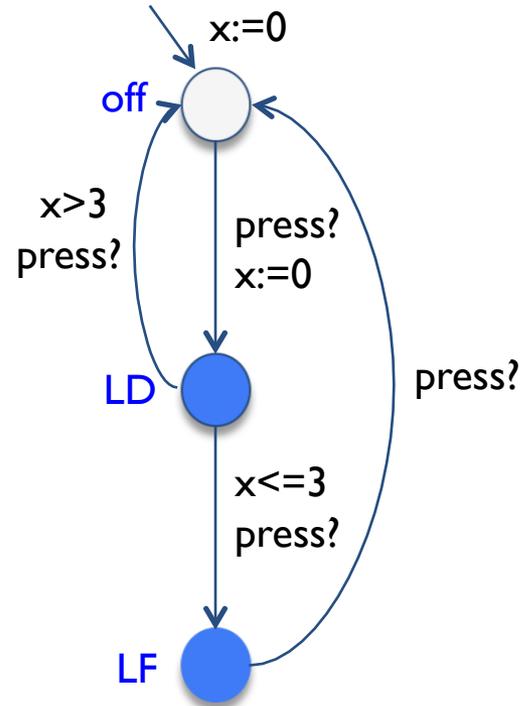


Propriété à vérifier: quand le train est dans la section du passage à niveau, la barrière est fermée.

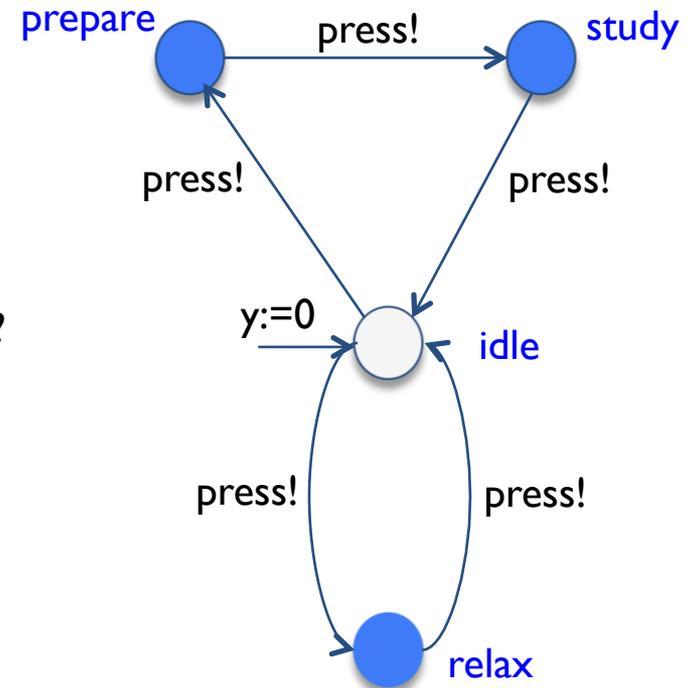
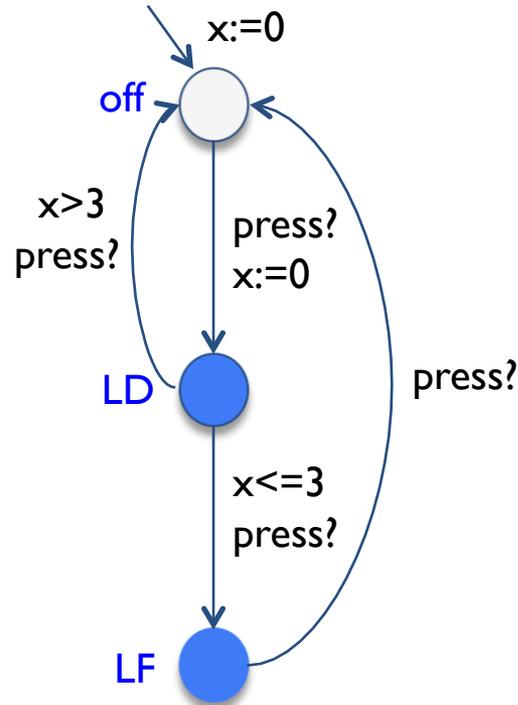
# Exercice



# Exercice

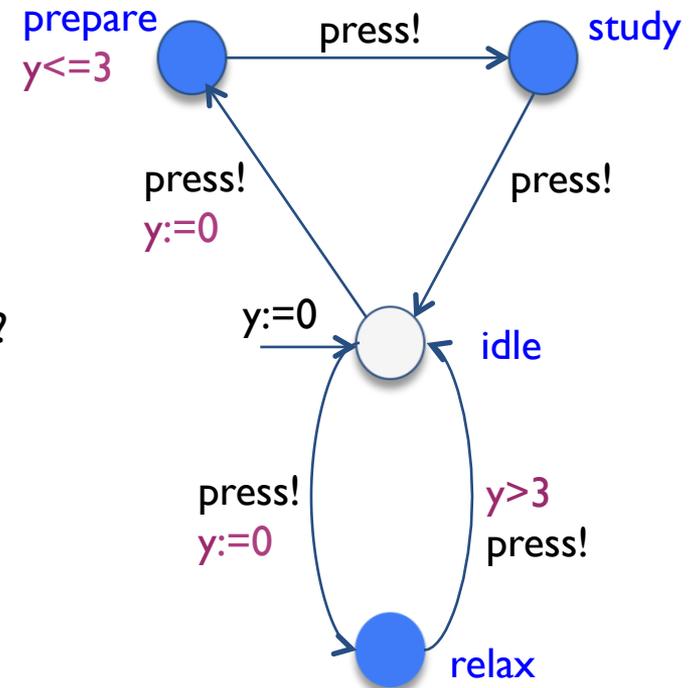
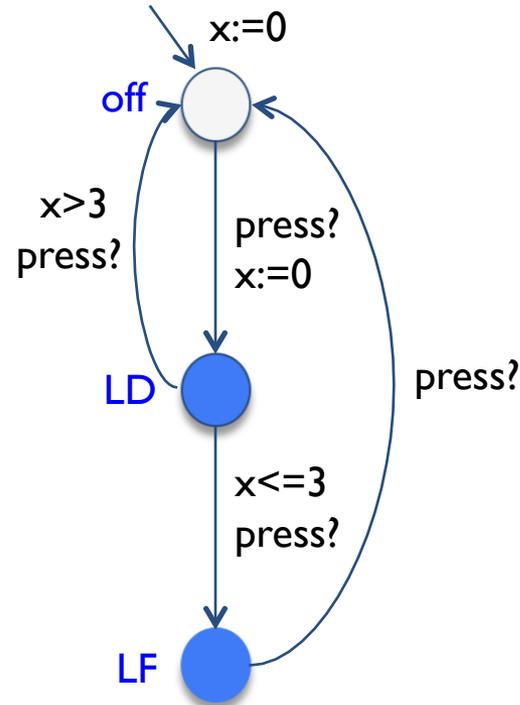


# Exercice



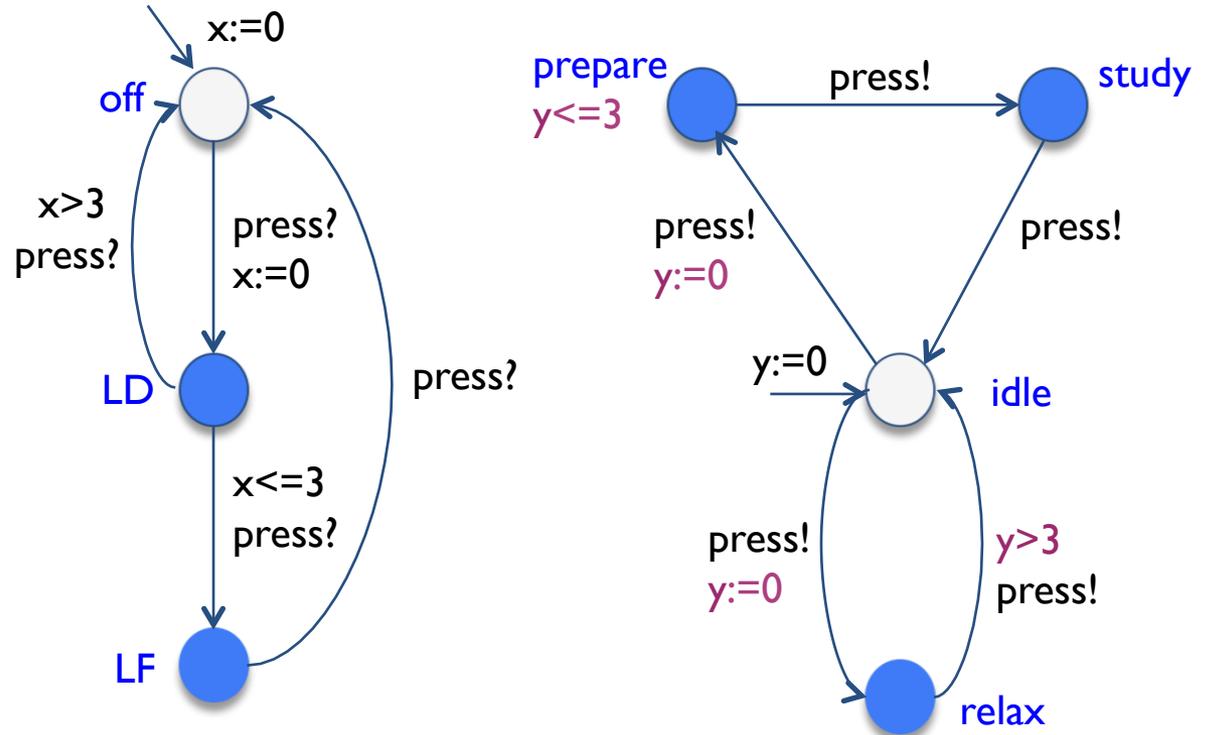
Ajouter une horloge pour que les deux composants soient compatible:  
study avec LF; relax avec LD

# Exercice



Ajouter une horloge pour que les deux composants soient compatible:  
study avec LF; relax avec LD

# Exercice



Composition ?

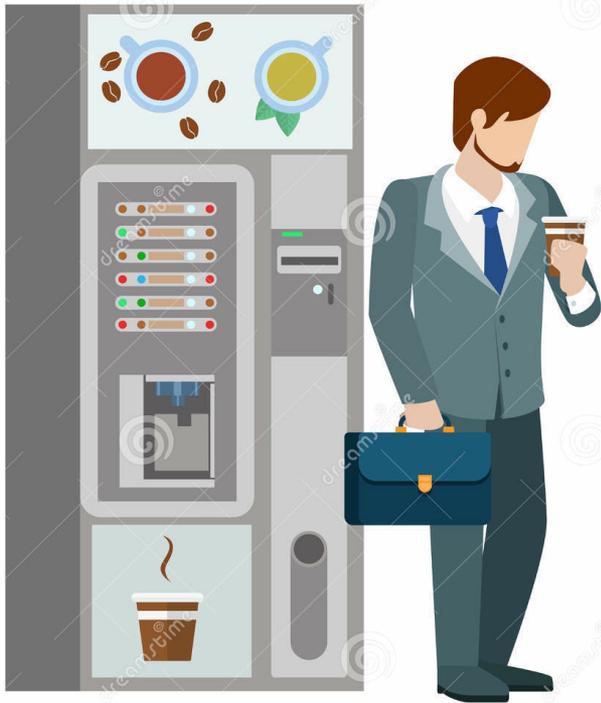
# Exercice

---



# Exercice

---

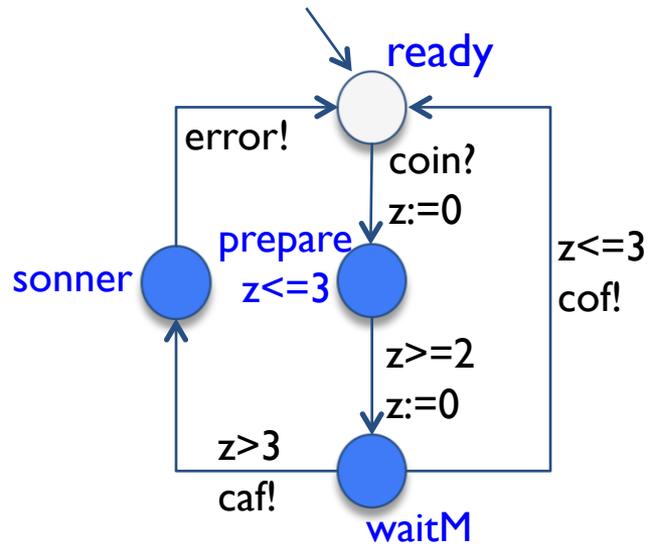


Deux composants (machine à café):

- Machine
  - ✧ recevoir coins
  - ✧ délivrer café (cas normal)
  - ✧ délivrer café (cas anormal)
  
- Personne
  - ✧ insérer coins
  - ✧ prendre café
  - ✧ faire pub

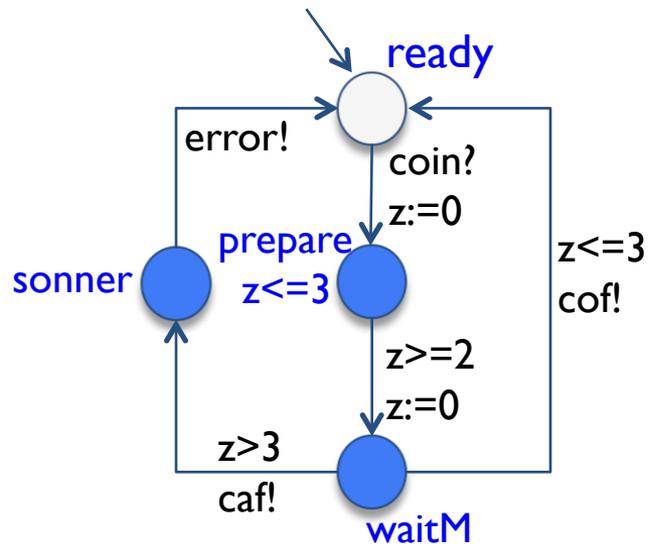
# Exercice

---

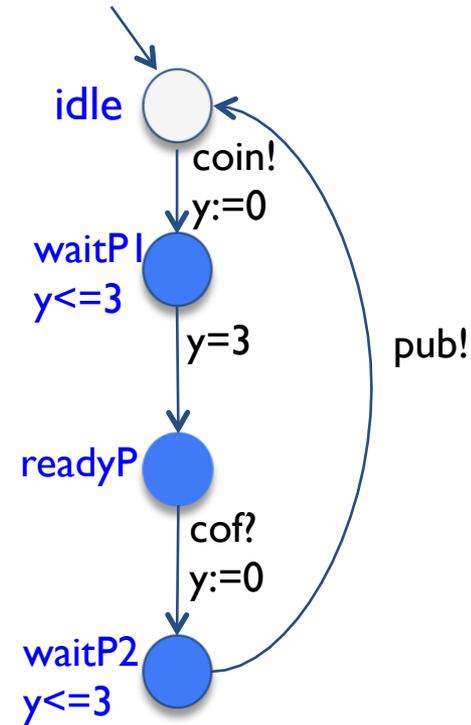


Machine

# Exercice

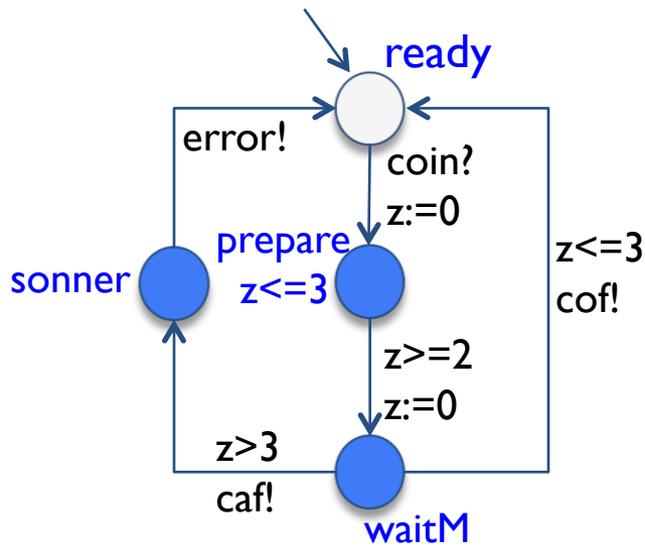


Machine

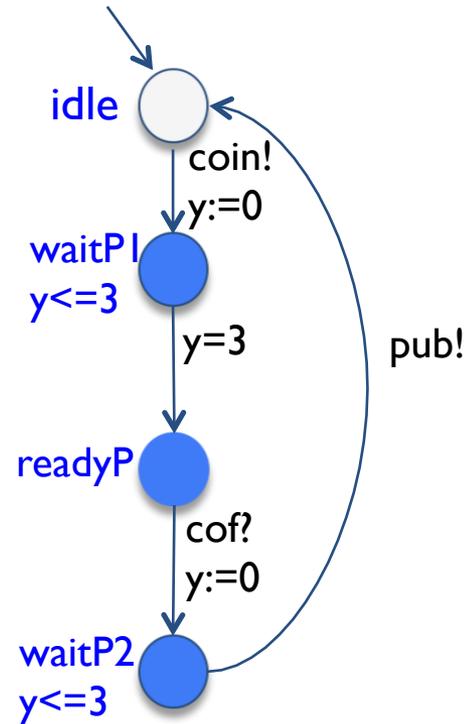


Person

# Exercice



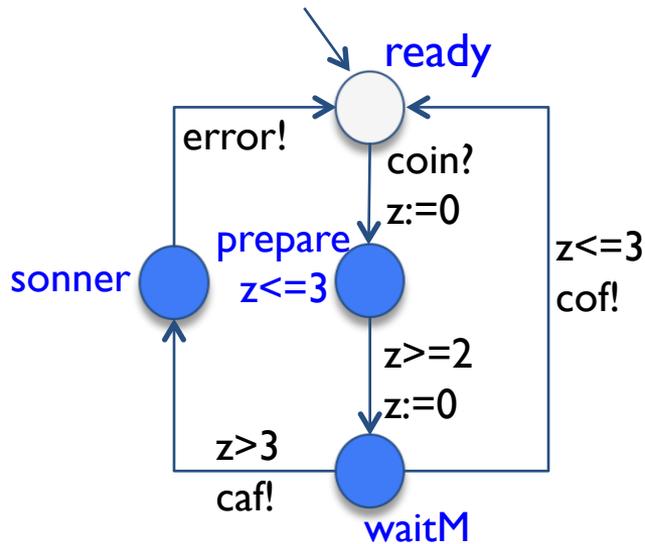
Machine



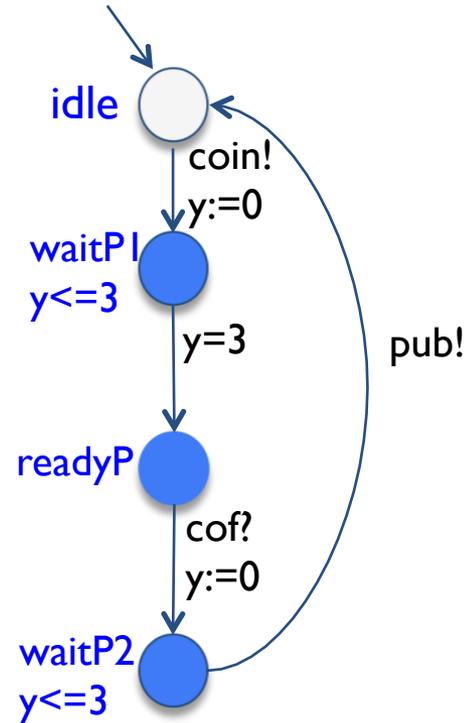
Person

Vérifier l'atteignabilité de tous les états: composition

# Exercice



Machine



Person

Comment faire pour que l'état sonner soit inatteignable?