Introduction aux automates temporisés

Bruno DENIS

Maître de Conférences à ENS de Cachan

Automates temporisés

- **Motivation**
  - Modélisation et analyse de systèmes réactifs à temps continu
  - Proposé par Alur et Dill en 1991

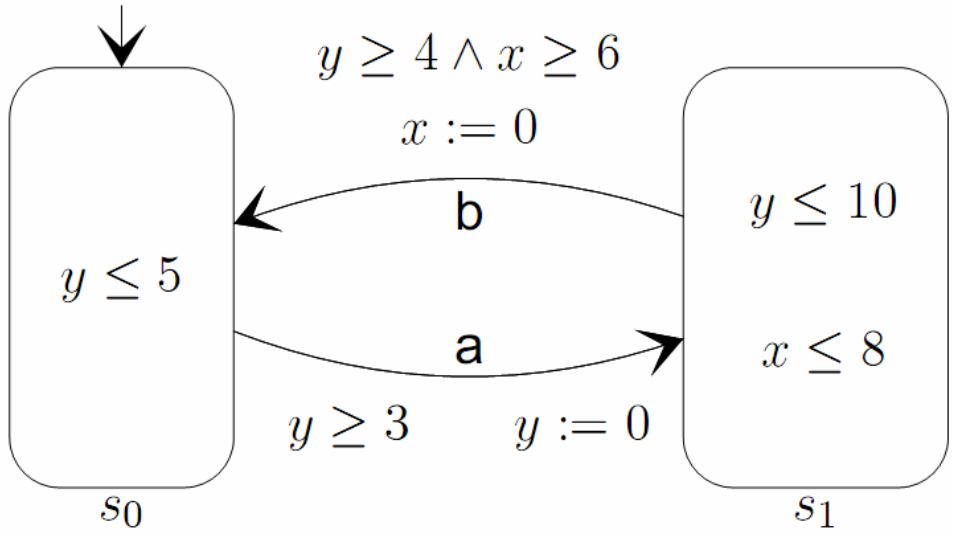
- **Qu'est-ce qu'un automate temporisé (Timed Automaton)**
  - Automate fini avec des **horloges à valeur dans $\mathbb{R}_+$**
    - qui fonctionnement simultanément et
    - qui peuvent être réinitialisées indépendamment
  - Chaque horloge mesure le temps écoulé depuis sa dernière initialisation
    - Mesure d'intervalles entre deux évènements

Présentation d'un automate temporisé

- Chaque automate possède un nombre fini de places (locations).
- Le franchissement d'une transition entre deux places est instantanée.
- Dans chaque place, le temps peut s'écouler: à tout instant, la valeur d'une horloge est le temps écoulé depuis sa dernière mise à 0.
- Les transitions entre places sont conditionnées par des contraintes sur les horloges, appelées gardes
- Les transitions entre places peuvent réinitialisé des horloges de l'automate (action).
- A chaque place est associée une contrainte sur les horloges, appelée invariant.

Exemple d'automate temporisé

- L'automate possède
  - deux places s_0 et s1 (s_0 initiale), et
  - deux horloges x et y
- s0 possède un invariant $y \leq 5$ tandis que s1 en possède deux, $y \leq 10$ et $y \leq 8$
- Deux transitions $s_0 \rightarrow s_1$ et $s_1 \rightarrow s_0$
- La transition $s_0 \rightarrow s_1$
  - est étiqueté par la lettre a,
  - est conditionnée par la garde $y \geq 3$,
  - assure l'action y := 0



Trajectoire d'un automate

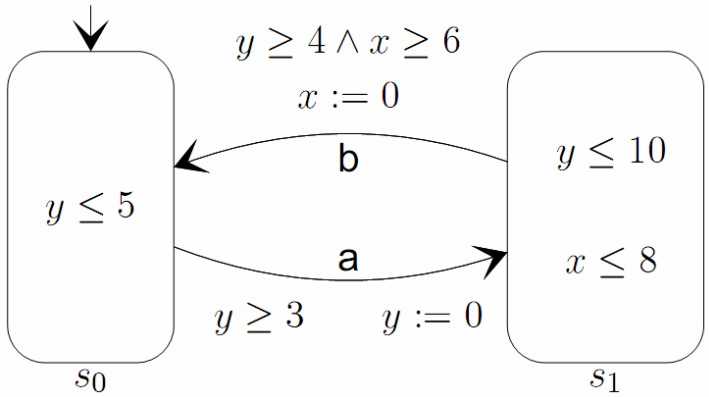
- **Une trajectoire est une exécution de l'automate**
  - Le franchissement d'une transition entre deux places est instantané
  - L'automate *peut* rester dans la place courante tant que son invariant est satisfait (une contrainte sur les horloges)
  - L'automate *doit* quitter la place courante avant que l'invariant ne soit violé
  - Pour franchir une transition il faut que sa garde soit vrai et que l'invariant de la place destination soit satisfait

- **Exemples**

$(s_0 , (0,0)) \xrightarrow[4]{a} (s_1 ,(4,0)) \xrightarrow[8]{b} (s_0 ,(0,4)) \ldots$

$(s_0 , (0,0)) \xrightarrow[5]{a} (s_1 ,(5,0)) \xrightarrow[8]{} (s_1 ,(8,3))$

État bloqué (deadlock)



Définition formelle des contraintes d'horloges

- Une horloge x est une variable à valeur dans $\mathbb{R}_+$, 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

$$\text{true} \mid x < c \mid x \leq c \mid x > c \mid x \geq c \mid \Phi_1 \wedge \Phi_2$$

où $x \in X$ est une horloge, $c \in \mathbb{Q}$ est un rationnel, et Φ_1, Φ_2 sont des contraintes d'horloge

Définition formelle

- **Un automate temporisé est $A = (\Sigma, S, s_0, X, I, T)$ avec**
  - Σ un ensemble fini d'étiquettes de transition
  - S un ensemble fini de places
  - $s_0 \in S$ une place initiale
  - X un ensemble fini de variable à valeur dans $\mathbb{R}_+$ (horloges)
  - $I : S \rightarrow C(X)$ pour définir les invariant de places
  - $T \subset S \times \Sigma \times C(X) \times 2^X \times S$ un ensemble de transitions
  - Chaque $e = < s, a, \Phi, \lambda, s' > \in T$ correspond à une transition entre la place s et la place s', gardée par la contrainte Φ, étiquetée par a, et qui réinitialise les variables $\lambda \subset X$

Composition d'automates temporisés

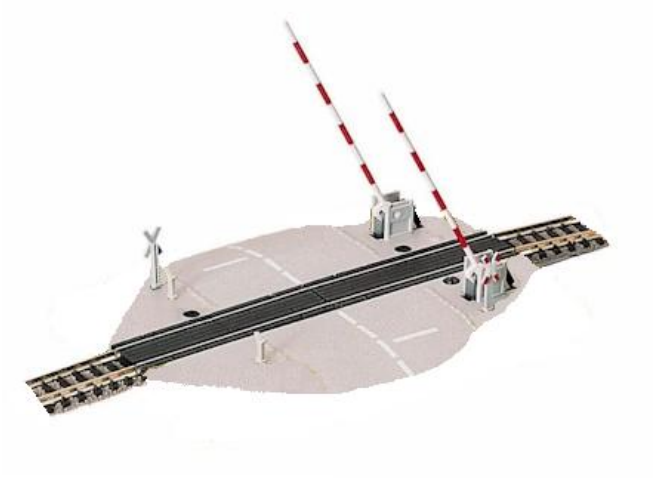
- Soient $A_1 = (\Sigma_1, S_1, s_{01}, X_1, I_1, T_1)$ et $A_2 = (\Sigma_2, S_1, s_{02}, X_2, I_2, T_2)$ avec $X_1 \cap X_2 = \varnothing$, alors $A_1 \,||\, A_2$ est l'automate temporisé

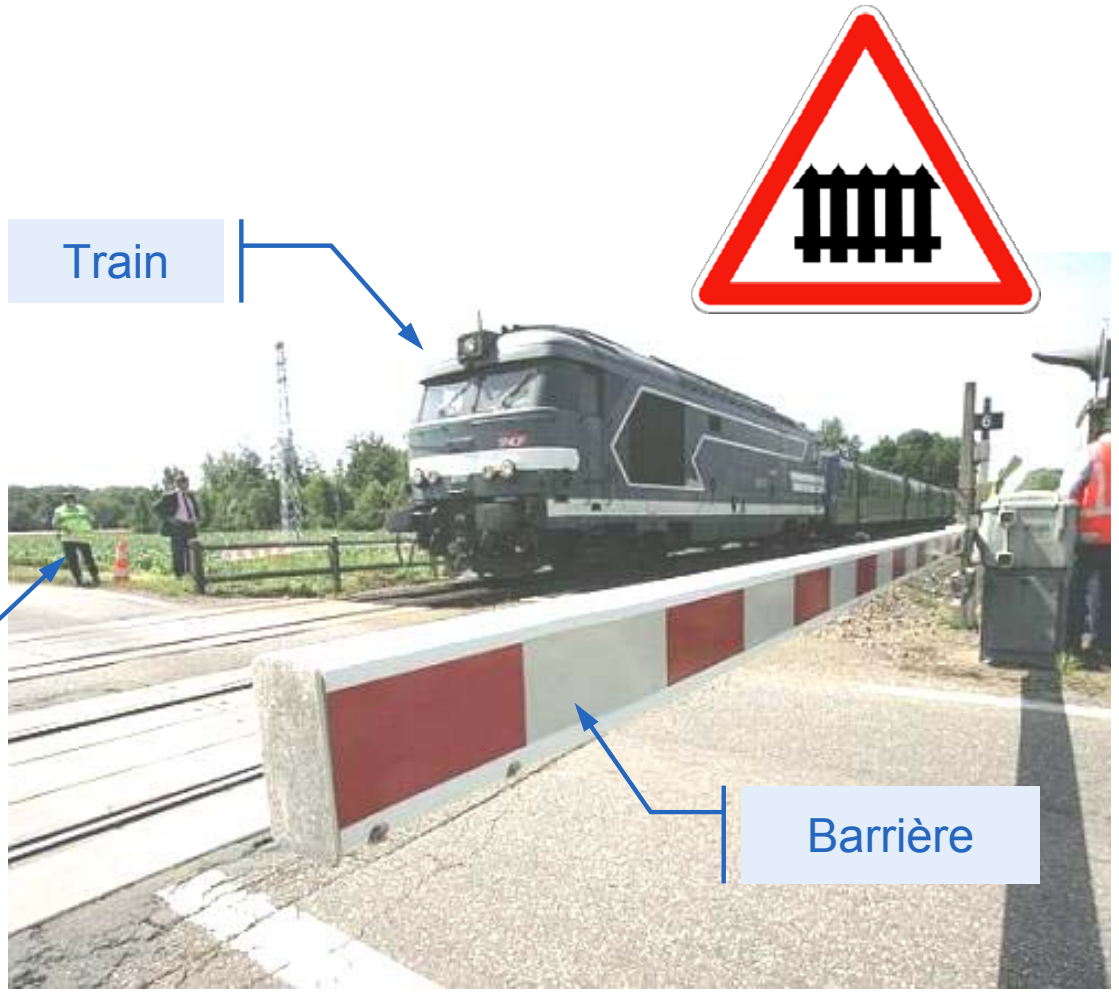
$$(\Sigma_1 \cup \Sigma_2, S_1 \times S_2, (s_{01}, s_{02}), X_1 \cup X_2, I, T) \text{ avec}$$

- $I(s_1, s_2) = I(s_1) \wedge I(s_2)$
- T est défini par
  1. Pour $a \in \Sigma_1 \cap \Sigma_2$
     $< s_1, a, \Phi_1, \lambda_1, s'_1 > \in T_1$ et $< s_2, a, \Phi_2, \lambda_2, s'_2 > \in T_2 \Rightarrow$
     $< (s_1, s_2), a, \Phi_1 \wedge \Phi_2, \lambda_1 \cup \lambda_2, (s'_1, s'_2) > \in T$
  2. Pour $a \in \Sigma_1 - \Sigma_2$
     $< s, a, \Phi, \lambda, s' > \in T_1$ et $t \in T_2 \Rightarrow < (s, t), a, \Phi, \lambda, (s', t) > \in T$
  3. symétriquement

Étude de cas : le passage à niveau

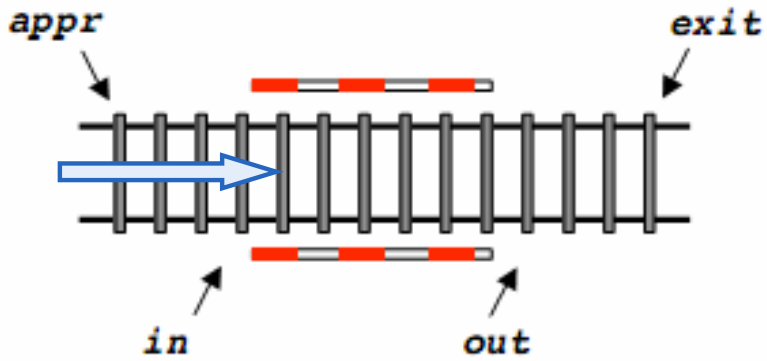
- Comportement temporel d'un système où 3 entités coopèrent





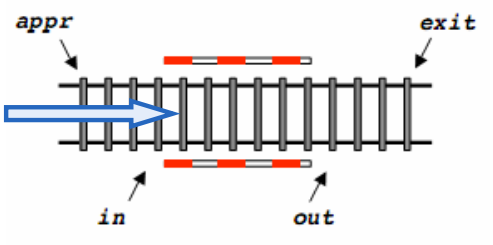
Étude de cas : le passage à niveau

Description du problème



- **Système faisant coopérer 3 composants :**
  - Le train
  - La barrière
  - Le contrôleur
  - Système = train || barrière || contrôleur
- **Propriétés à vérifier :**
  - Quand le train est dans la section du passage à niveau, la barrière est fermée (non temporisée);
  - La barrière ne reste jamais fermée plus de 10 mn d'affilée (temporisée).

Étude de cas : le passage à niveau



Barrière

- Lorsque la barrière reçoit le signal « baisser », elle atteint la position basse **en moins d'1 mn**
- Lorsque la barrière reçoit le signal « lever », elle met **entre 1 et 2 mn** pour atteindre la position haute.
- Si elle ne reçoit pas de signal, la barrière peut rester indéfiniment en position haute ou basse.

Train

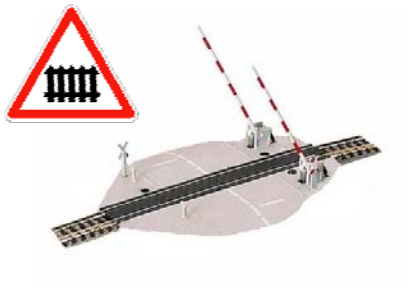
- Le train signale son approche **au moins 2 mn** avant d'entrer dans la section gardée ;
- Il ne s'écoule **pas plus de 5 mn** entre l'approche et la sortie de zone contrôlée

Contrôleur

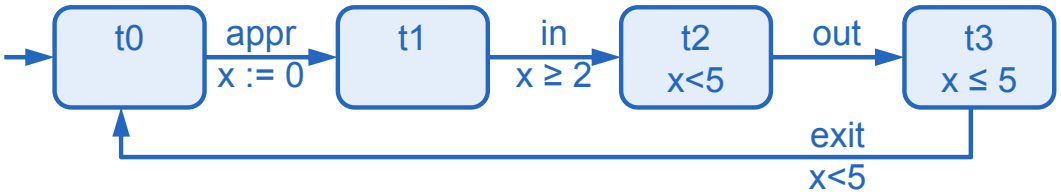
- A partir du moment où le train signale son approche, il s'écoule **exactement 1 mn** avant que le contrôleur envoie le signal « baisser »
- Lorsque il reçoit le signal « exit », le contrôleur met **moins d'1 mn** pour envoyer le signal « lever ».

Étude de cas : le passage à niveau

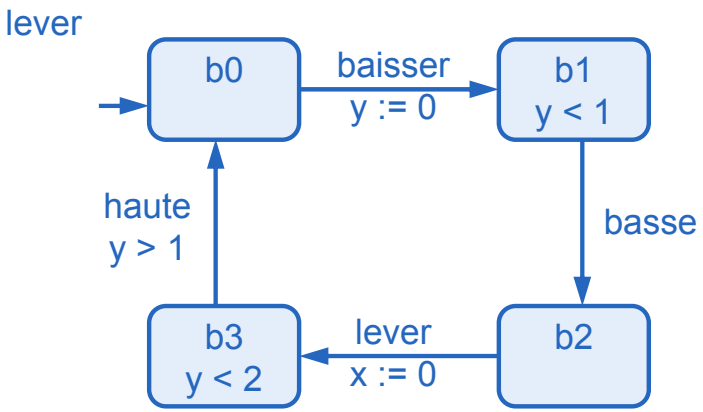
- **Modèle du système : Système = Train || Barrière || Contrôleur**



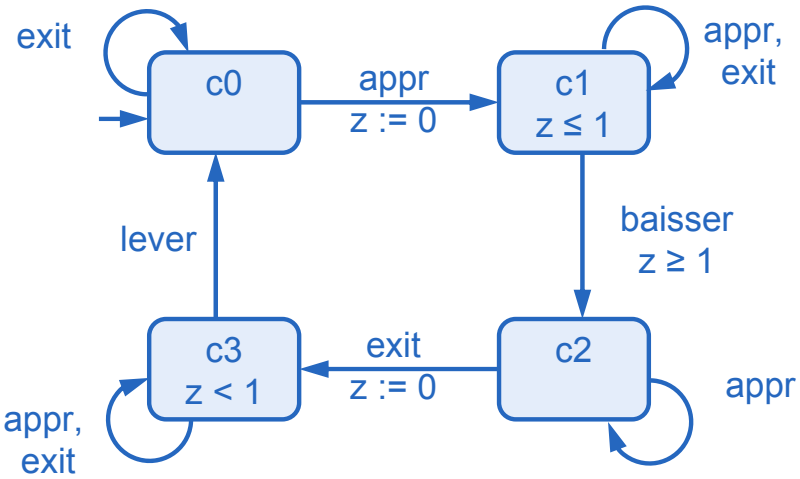
Train



Barrière



Contrôleur



Étude de cas : le passage à niveau