

# Réseaux de Petri T-temporels et automates temporisés

Didier LIME

`Didier.Lime@irccyn.ec-nantes.fr`

Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN)

CNRS UMR 6597

Ecole Centrale

1 rue de la Noë B.P. 92101

44321 Nantes cedex 03

Equipe Systèmes Temps-réel

# Introduction

- ▶ Vérifier des propriétés **temporelles quantitatives** sur des réseaux de Petri T-temporels

# Introduction

- ▶ Vérifier des propriétés **temporelles quantitatives** sur des réseaux de Petri T-temporels
- ▶ Graphe des classes + observateur

# Introduction

- ▶ Vérifier des propriétés **temporelles quantitatives** sur des réseaux de Petri T-temporels
- ▶ Graphe des classes + observateur
- ▶ Construire un automate temporisé **temporellement bisimilaire** au RdPT et vérifier des propriétés avec UPPAAL ou KRONOS par exemple

# Introduction

- ▶ Vérifier des propriétés **temporelles quantitatives** sur des réseaux de Petri T-temporels
- ▶ Graphe des classes + observateur
- ▶ Construire un automate temporisé **temporellement bisimilaire** au RdPT et vérifier des propriétés avec UPPAAL ou KRONOS par exemple
- ▶ Trois approches :
  - Traduction structurelle
  - Calcul par le graphe des régions
  - Calcul par le graphe des classes

# Plan

- ▶ Définitions
  - Réseaux de Petri T-temporels
  - Automates temporisés
- ▶ Graphe des classes d'états
- ▶ Des réseaux de Petri T-temporels vers les automates temporisés
  - Traduction structurelle
  - Graphe des régions
  - Automate des classes d'états
    - Graphe des classes étendues
    - Automate temporisé des classes
    - Résultats

# Réseaux de Petri T-temporels - Définition

**Définition 1 (Réseau de Petri T-temporel)** *Un réseau de Petri T-temporel est un 7-uplet*

$\mathcal{T} = (P, T, \bullet(\cdot), (\cdot)^\bullet, \alpha, \beta, M_0)$ , où

- ▶  $P = \{p_1, p_2, \dots, p_m\}$  est un ensemble fini et non vide de **places**,
- ▶  $T = \{t_1, t_2, \dots, t_n\}$  est un ensemble fini et non vide de **transitions** ( $T \cap P = \emptyset$ ),
- ▶  $\bullet(\cdot) \in (\mathbb{N}^P)^T$  est la **fonction d'incidence arrière**,
- ▶  $(\cdot)^\bullet \in (\mathbb{N}^P)^T$  est la **fonction d'incidence avant**,
- ▶  $M_0 \in \mathbb{N}^P$  est le **marquage initial** du réseau,
- ▶  $\alpha \in (\mathbb{Q}^+)^T$  et  $\beta \in (\mathbb{Q}^+ \cup \{\infty\})^T$  sont les fonctions donnant, pour chaque transition, respectivement son instant de tir **au plus tôt** et **au plus tard** ( $\alpha \leq \beta$ ).

# Réseaux de Petri T-temporels - Sémantique (1/2)

**Définition 2 (Sémantique d'un RdPT)** *La sémantique d'un RdPT  $\mathcal{T}$  est donnée sous la forme d'un STT  $\mathcal{S}_{\mathcal{T}} = (Q, q_0, \rightarrow)$  tel que*

- ▶  $Q = \mathbb{N}^P \times (\mathbb{R}^+)^T$
- ▶  $q_0 = (M_0, \bar{0})$
- ▶  $\rightarrow \in Q \times (T \cup \mathbb{R}) \times Q$  est la relation de transition comportant une relation de transition continue et une relation de transition discrète

# Réseaux de Petri T-temporels - Sémantique (2/2)

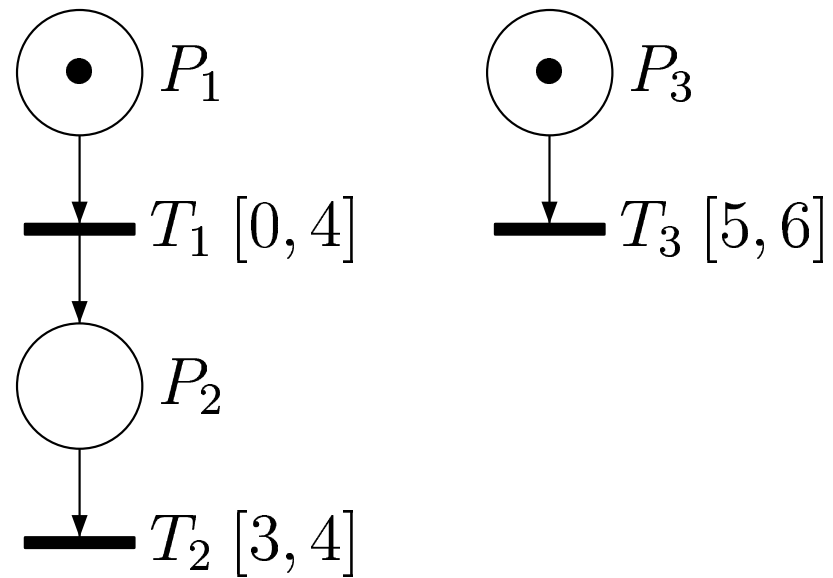
- ▶ La relation de transition **continue** est définie  $\forall d \in \mathbb{R}^+$  par :

$$(M, \nu) \xrightarrow{d} (M, \nu') \text{ ssi } \begin{cases} \nu' = \nu + d, \\ \forall t_k \in T, M \geq \bullet t_k \Rightarrow \nu'(t_k) \leq \beta(t_k) \end{cases}$$

- ▶ La relation de transition **discrète** est définie  $\forall t_i \in T$  par :

$$(M, \nu) \xrightarrow{t_i} (M', \nu') \text{ ssi } \begin{cases} M \geq \bullet t_i, \\ M' = M - \bullet t_i + t_i \bullet, \\ \alpha(t_i) \leq \nu(t_i) \leq \beta(t_i), \\ \forall t_k, \nu(t_k)' = \begin{cases} 0 & \text{si } \uparrow \text{enabled}(t_k, M, t_i) \\ \nu(t_k) & \text{sinon} \end{cases} \end{cases}$$

# RdPT - Exemple



# Automate temporisé - Définition

**Définition 3 (Automate temporisé)** [HNSY94] Un **automate temporisé** est un 6-uplet  $(L, l_0, X, A, E, Inv)$  où

- ▶  $L$  est un ensemble fini de **localités**,
- ▶  $l_0$  est la **localité initiale**,
- ▶  $X$  est un ensemble fini d'**horloges** à valeurs réelles positives,
- ▶  $A$  est un ensemble fini d'**actions**,
- ▶  $E \subset L \times \mathcal{C}(X) \times A \times 2^X \times L$  est un ensemble fini d'arcs. Soit  $e = (l, \delta, \alpha, R, \rho, l') \in E$ .  $e$  est l'arc reliant la localité  $l$  à la localité  $l'$ , avec la **garde**  $\delta$ , l'action  $\alpha$ , l'ensemble d'horloges à réinitialiser  $R$  et la fonction de renommage  $\rho$ .
- ▶  $Inv \in \mathcal{C}(X)^L$  associe un **invariant** à chaque localité

# Automate temporisé - Sémantique

**Définition 4 (Sémantique d'un AT)** La sémantique d'un AT  $H$  est définie comme un STT  $\mathcal{S}_H = (Q, Q_0, \rightarrow)$  où  $Q = L \times (\mathbb{R}^+)^X$ ,  $Q_0 = (l_0, \bar{0})$  est l'état initial et  $\rightarrow$  est définie, pour  $a \in A$  et  $t \in \mathbb{R}_+$ , par

► les transitions **discrètes** :  $(l, \nu) \xrightarrow{a} (l', \nu')$  ssi

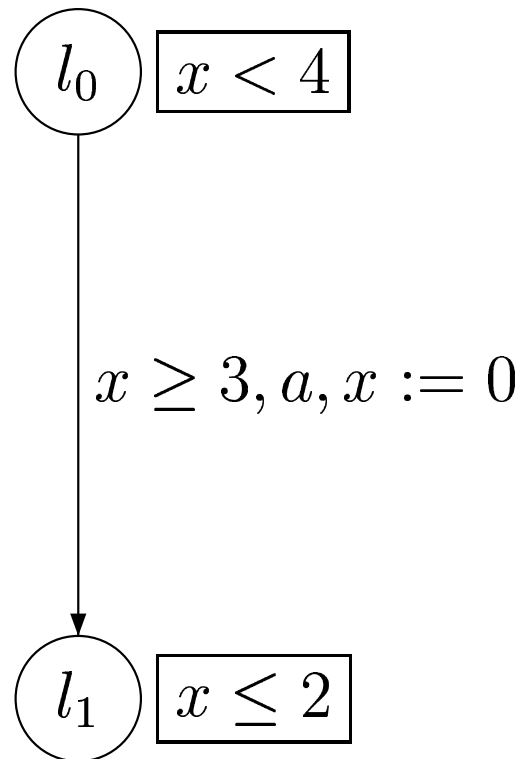
$\exists (l, \delta, a, R, \rho, l') \in E$  tel que

$$\begin{cases} \delta(\nu) = true, \\ \nu' = \nu[R \leftarrow 0][\rho], \\ Inv(l')(\nu') = true \end{cases}$$

► les transitions **continues** :  $(l, \nu) \xrightarrow{t} (l, \nu')$  ssi

$$\begin{cases} \nu' = \nu + t, \\ \forall t' \in [0, t], Inv(l)(\nu + t') = true \end{cases}$$

# AT - Exemple



# Plan

- ▶ Définitions
  - Réseaux de Petri T-temporels
  - Automates temporisés
- ▶ **Graphe des classes d'états**
- ▶ Des réseaux de Petri T-temporels vers les automates temporisés
  - Traduction structurelle
  - Graphe des régions
  - Automate des classes d'états
    - Graphe des classes étendues
    - Automate temporisé des classes
    - Résultats

# Graphe des classes d'états - Classe d'états

**Définition 5 (Classe d'états)** Une **classe d'états**  $C$ , d'un RdPT, est un couple  $(M, D)$  où  $M$  est un marquage du réseau et  $D$  un ensemble d'inéquations appelé **domaine de tir**.

Les inéquations de  $D$  sont de deux types [BD91]

$$\begin{cases} \alpha_i \leq \theta_i \leq \beta_i \ (\forall i \text{ tel que } t_i \text{ est sensibilisée}), \\ -\gamma_{kj} \leq \theta_j - \theta_k \leq \gamma_{jk}, \ \forall j, k \text{ tel que } j \neq k \text{ et } (t_j, t_k) \in \text{enabled}(M)^2 \end{cases}$$

$\theta_i$  est l'instant de tir de la transition sensibilisée  $t_i$  **relativement** à l'instant d'entrée dans la classe.

# Graphe des classes d'états - Classe suivante

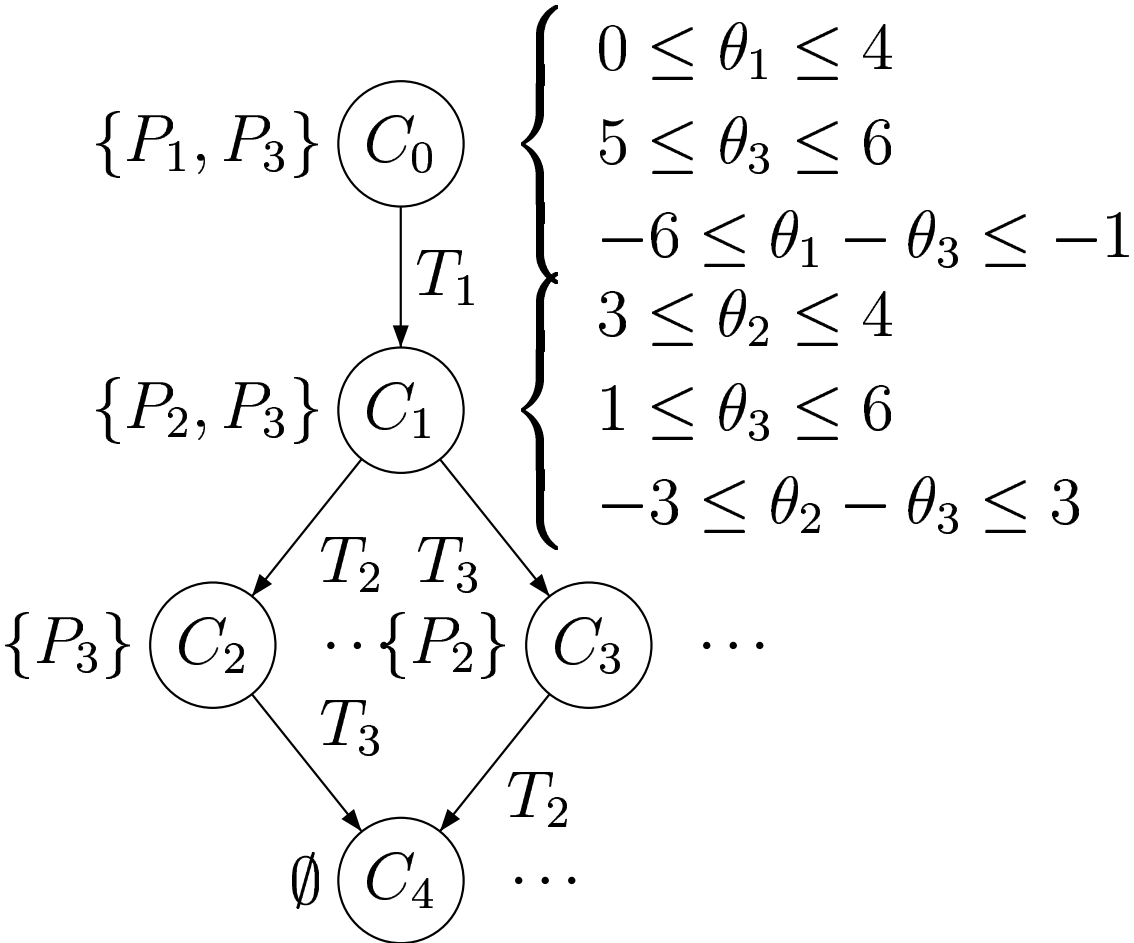
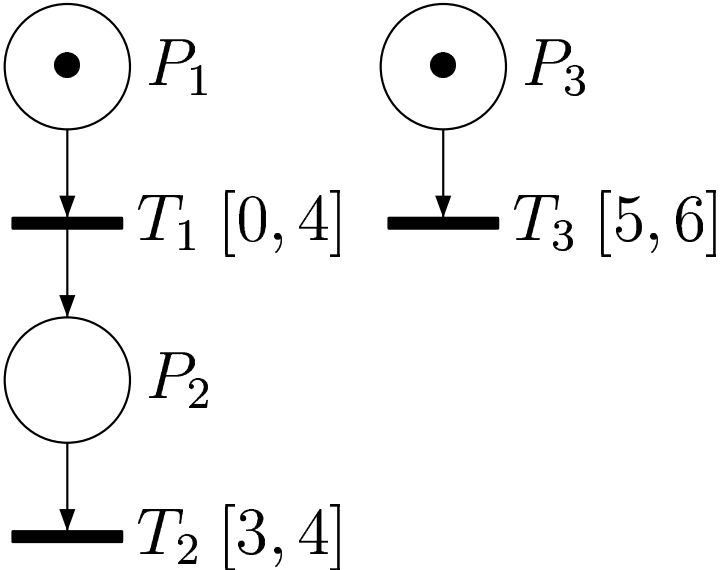
Etant donnée une classe  $C = (M, D)$  et une transition **franchissable**  $t_f$ , la classe  $C' = (M', D')$  obtenue par le tir de  $t_f$  depuis  $C$  est donnée par

- ▶  $M' = M - \bullet t_f + t_f \bullet$
- ▶ Le calcul de  $D'$  requiert les étapes suivantes
  1. changements de variables  $\forall j, \theta_j = \theta_f + \theta'_j$ ,
  2. élimination de toutes les variables relatives à des transitions désensibilisées par le tir de  $t_f$ ,
  3. ajout d'inéquations relatives aux transitions nouvellement sensibilisées

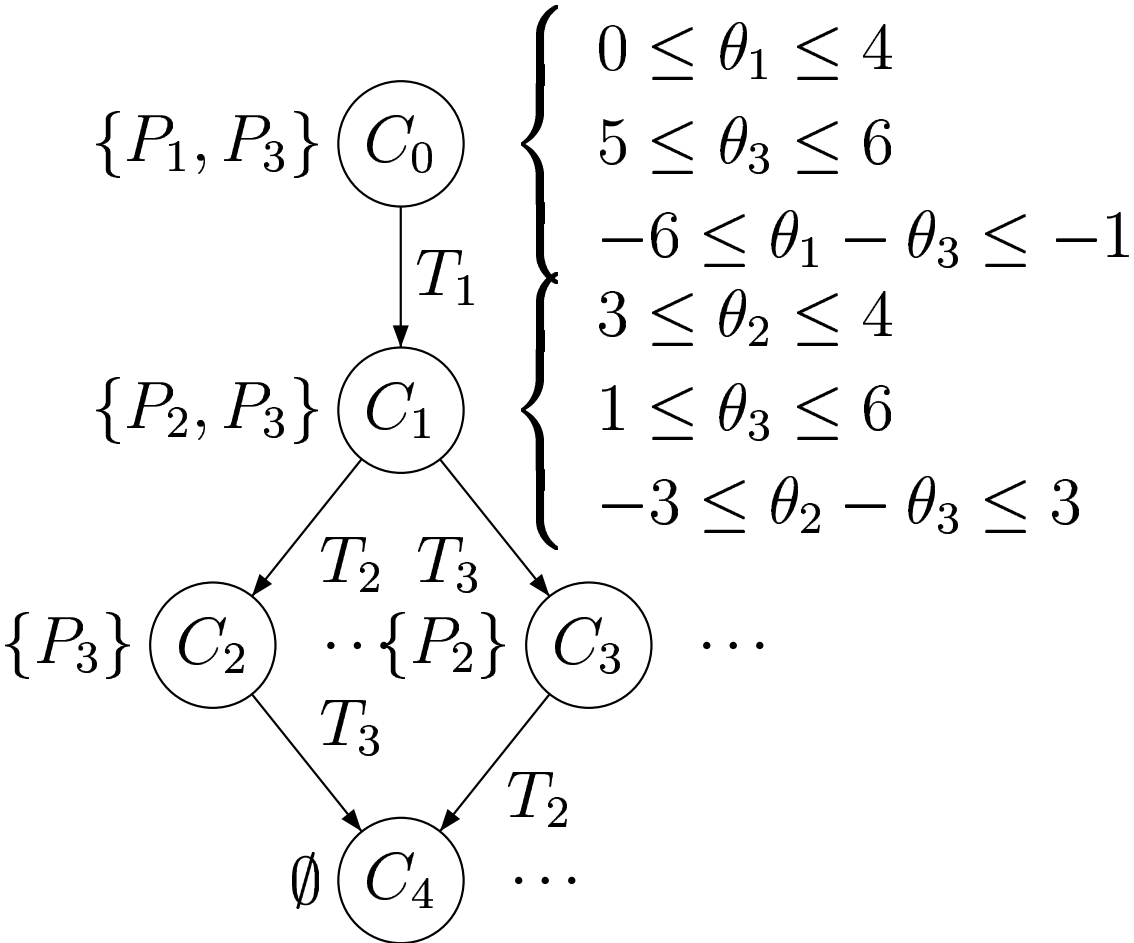
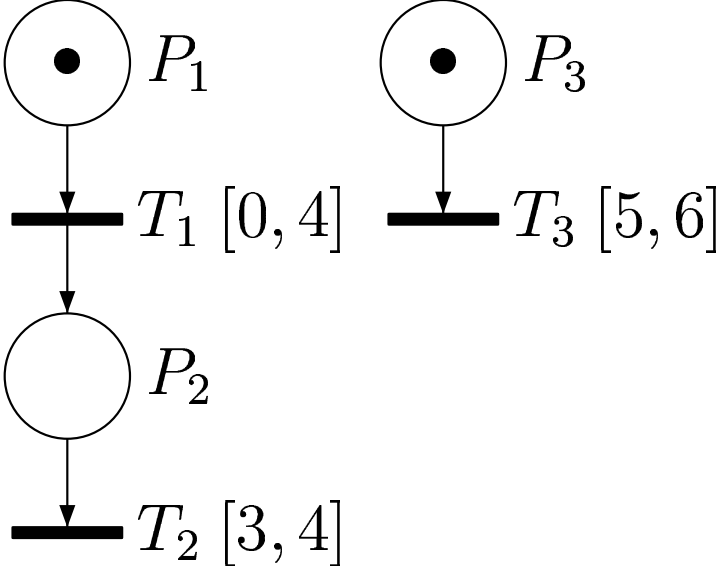
$$\forall t_k \in \uparrow \text{enabled}(M, t_f), \alpha(t_k) \leq \theta'_k \leq \beta(t_k).$$

4. détermination de la forme canonique  $D'^*$

# Graphe des classes d'états - Exemple

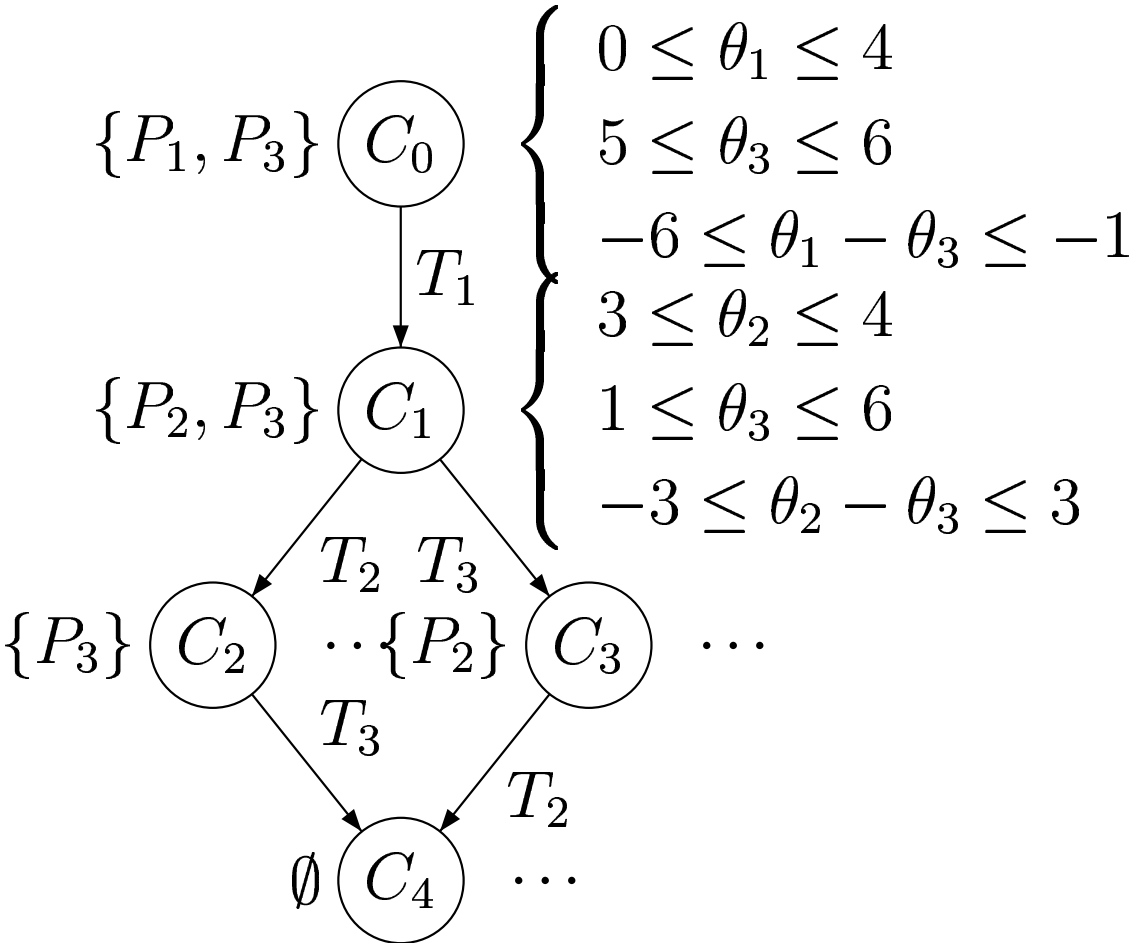
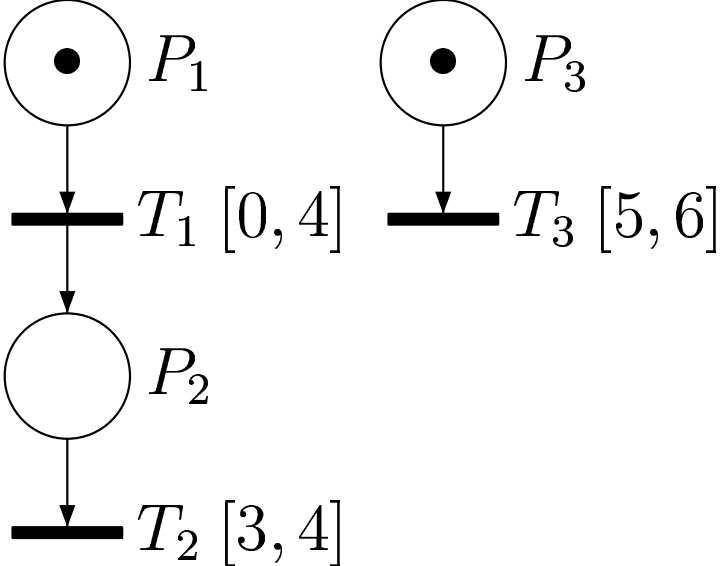


# Graphe des classes d'états - Exemple



Langage **temporel** ?

# Graphe des classes d'états - Exemple



Langage **temporel** ?  $\Rightarrow$  observateurs

# Plan

- ▶ Définitions
  - Réseaux de Petri T-temporels
  - Automates temporisés
- ▶ Graphe des classes d'états
- ▶ **Des réseaux de Petri T-temporels vers les automates temporisés**
  - Traduction structurelle
  - Graphe des régions
  - Automate des classes d'états
    - Graphe des classes étendues
    - Automate temporisé des classes
    - Résultats

# Traduction structurelle (1/3)

- ▶ Franck Cassez et Olivier H. Roux
- ▶  $\mathcal{T} = (P, T, \bullet(\cdot), (\cdot)^\bullet, M_0, (\alpha, \beta))$  est un réseau de Petri T-temporel, avec  $P = \{p_1, \dots, p_m\}$  et  $T = \{t_1, \dots, t_n\}$ .
- ▶ L'automate temporisé obtenu **structurellement** est le **produit synchronisé** :  $\Delta(\mathcal{T}) = (SU \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n)_f$  où  $SU$  est le **superviseur**, et  $\mathcal{A}_i$  un automate temporisé **pour chaque transition**  $t_i$  de  $T$ .
- ▶ Une horloge pour chaque automate  $\mathcal{A}_i$
- ▶ Les états de l'automate  $\mathcal{A}_i$  donnent l'état de la transition  $t_i$
- ▶ Complexité du calcul faible (linéaire)

# Traduction structurelle (2/3)

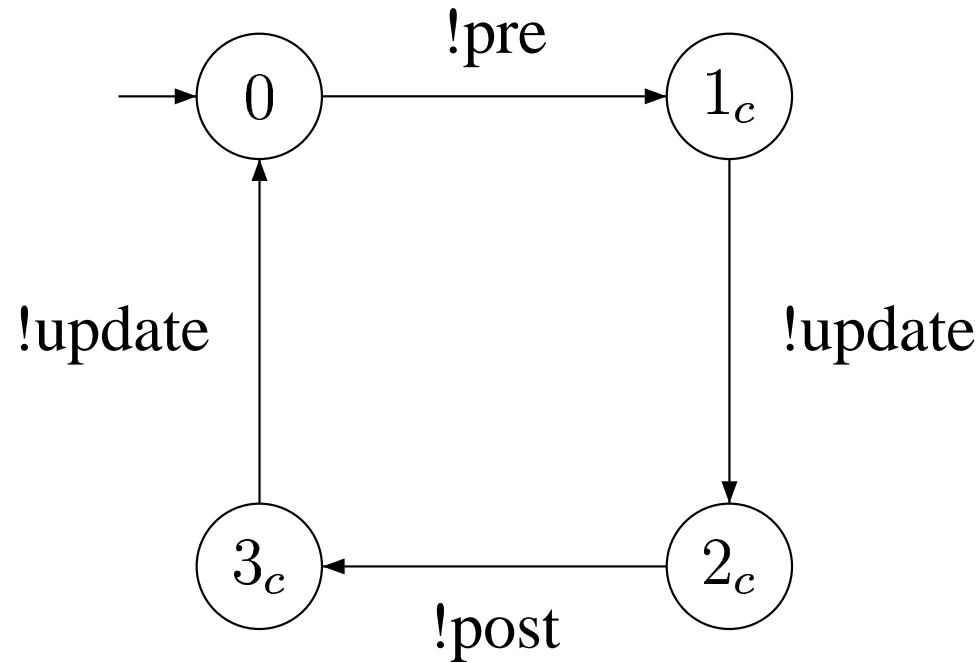


Figure 1: Automate du superviseur  $SU$

# Traduction structurelle (3/3)

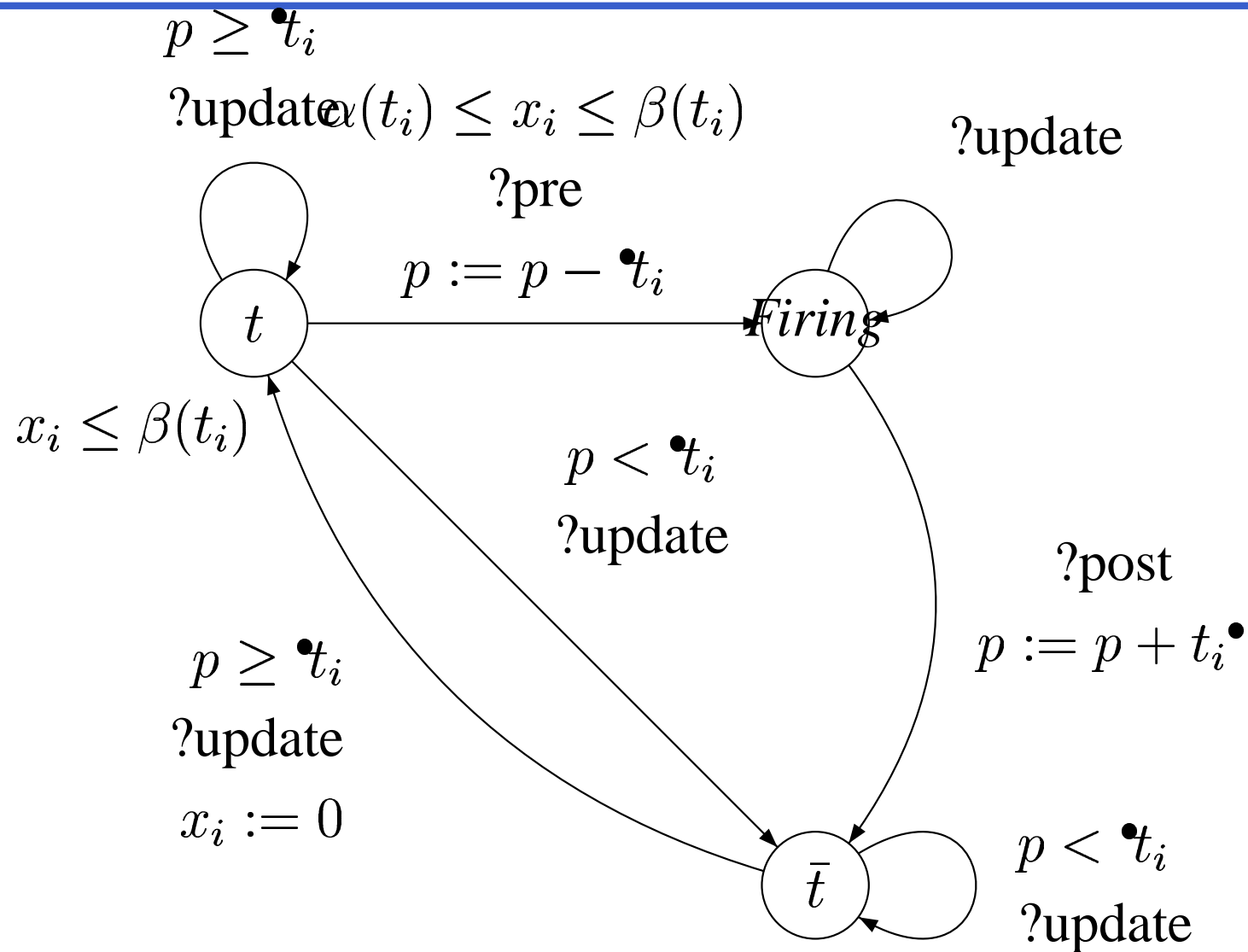
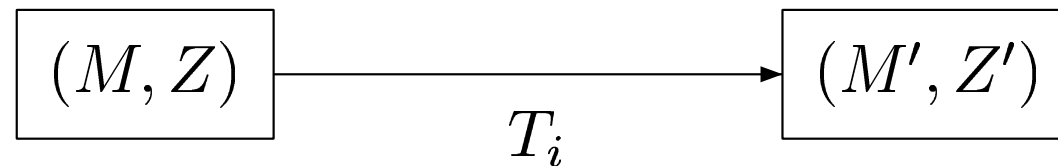


Figure 2: Automate  $\mathcal{A}_i$  correspondant à la transition  $t_i$

# Graphe des Régions (1/2)

- ▶ Algorithme de calcul en avant de l'espace d'états d'un automate temporisé (UPPAAL, KRONOS).



$$\alpha_i \leq x_i \leq \beta_i$$

$$Z' = \text{reset}_K \left( \vec{Z} \cap I(M) \cap x_i \geq \alpha_i \right)$$

- ▶ SAVA et ALLA 2001: traduction en automate temporisé (synthèse de la commande)
- ▶ Convergence pour les réseaux de Petri T-temporels bornés avec dates de tir spécifiées par des rationnels.

# Graphe des Régions (2/2)

- ▶ Guillaume Gardey, Olivier H. Roux et Olivier Roux
- ▶ Implémentation: DBM.
- ▶ BOUYER (2002), problème sur-approximation pour les automates temporisés.
- ▶ Réseaux de Petri T-temporels:

$(\alpha, \beta) \in Q \times Q \rightarrow$  graphe des régions exact

$(\alpha, \beta) \in Q \times Q \cup \{\infty\} \rightarrow$  sur-approximation

mais ensemble des marquages accessibles exact!

- ▶ Analyse (au-vol) de l'accessibilité de marquage.
- ▶ Bisimulation.

# Plan

- ▶ Définitions
  - Réseaux de Petri T-temporels
  - Automates temporisés
- ▶ Graphe des classes d'états
- ▶ Des réseaux de Petri T-temporels vers les automates temporisés
  - Traduction structurelle
  - Graphe des régions
  - **Automate des classes d'états**
    - Graphe des classes étendues
    - Automate temporisé des classes
    - Résultats

# Classe d'états étendue

**Définition 6 (Classe d'états étendue)** Une **classe d'état étendue** est un 4-uplet  $(M, D, \chi, \text{trans})$ , où  $M$  est un marquage,  $D$  un domaine de tir,  $\chi$  un ensemble d'horloges à valeur réelles et  $\text{trans} \in (2^T)^X$  associe des ensembles de transitions aux horloges.

# Graphe des classes d'états étendues

A chaque classe calculée, on effectue les étapes supplémentaires suivantes

1. pour chaque horloge  $x$  de  $\chi$ , les transitions désensibilisées sont enlevées de  $trans(x)$ ,
2. les horloges dont l'image par  $trans$  est vide sont enlevées de  $\chi$ ,
3. s'il y a des transitions nouvellement sensibilisées, deux cas de figures se présentent :
  - ▶ il existe une horloge  $x$  dont la valeur est 0. Alors, on se contente d'ajouter les transitions nouvellement sensibilisées à  $trans(x)$ ,
  - ▶ Une telle horloge n'existe pas. Alors on crée une nouvelle horloge  $x_i$ .  $i = \min\{i \in \mathbb{N} | x_i \notin \chi\}$ . On ajoute  $x_i$  à  $\chi$  et  $trans(x_i)$  est l'ensemble des transitions nouvellement sensibilisées

# Critère de convergence

**Définition 7 (Clock-similarity)** Deux classes d'états étendues  $C = (M, D, \chi, trans)$  et  $C' = (M', D', \chi', trans')$  sont **clock-similar**, ce qui est noté  $C \approx C'$ , ssi elles ont le même marquage, le même nombre d'horloges et que ces horloges sont associées aux mêmes transitions :

$$C \approx C' \Leftrightarrow \begin{cases} M = M', \\ |\chi| = |\chi'|, \\ \forall x \in \chi, \exists x' \in \chi', trans(x) = trans'(x'). \end{cases}$$

- ▶ **Clock-similarity** est le critère de convergence pour le graphe des classes étendues
- ▶ si en plus on a une **inclusion** de domaines de tir alors l'algorithme s'arrête, sinon on poursuit l'exploration des fils hors de l'intersection

# Automate temporisé des classes d'états (1/2)

**Définition 8 (Automate temporisé des classes d'états)** *L'automate temporisé des classes d'états  $\Delta(\mathcal{T}) = (L, l_0, X, A, E, Inv)$  est défini à partir du graphe des classes étendues par :*

- ▶  *$L$ , l'ensemble des localités, est l'ensemble des classes d'états étendues  $C^{ext}$ ,*
- ▶  *$l_0$  est la classe initiale  $(M_0, D_0, \chi_0, trans_0)$ ,*
- ▶  *$X = \bigcup_{(M,D,\chi,trans) \in C^{ext}} \chi$*
- ▶  *$A = T$  est l'ensemble des transitions*

# Automate temporisé des classes d'états (2/2)

- $E$  est l'ensemble des arcs défini par

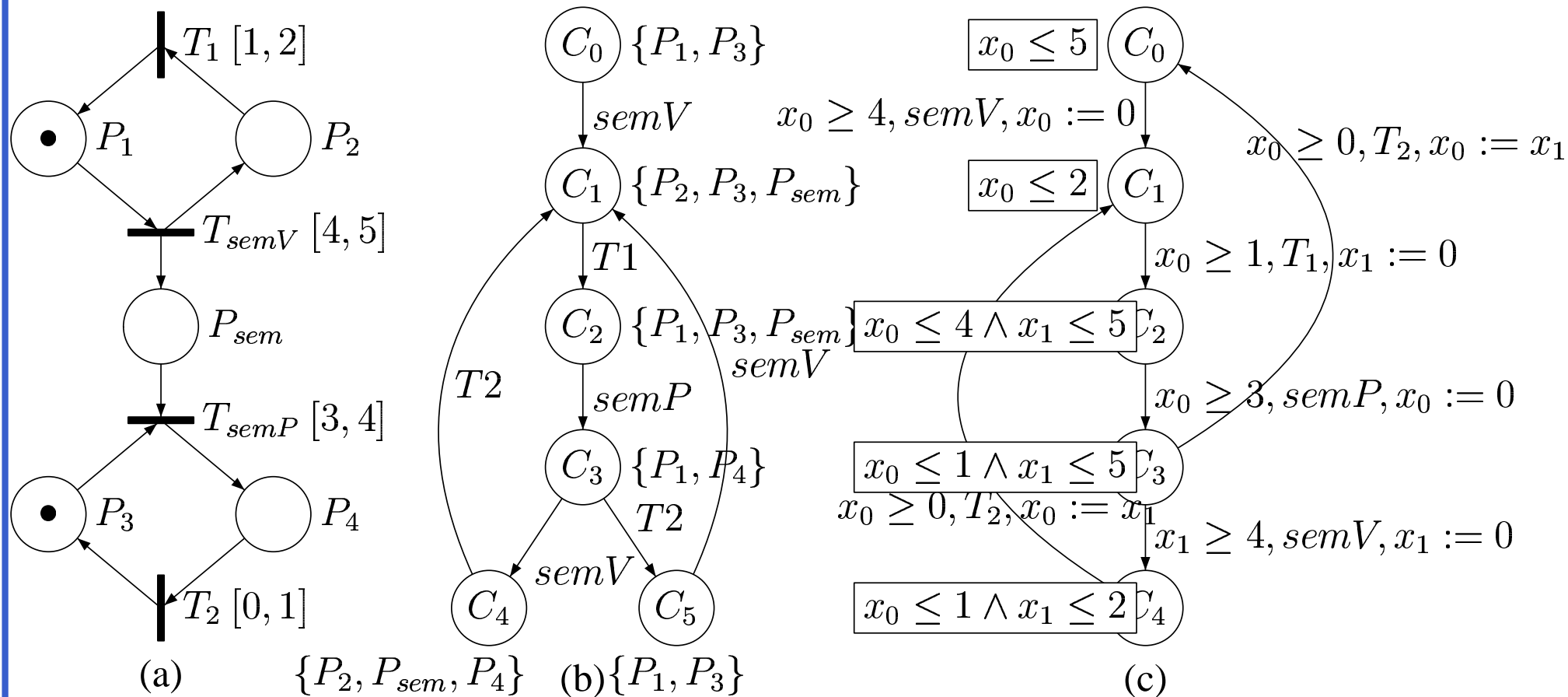
$$\forall C_i = (M_i, D_i, \chi_i, trans_i), C_j = (M_j, D_j, \chi_j, trans_j) \in C^{ext},$$

si  $\exists C_i \xrightarrow[t]{ext} C_j$  alors

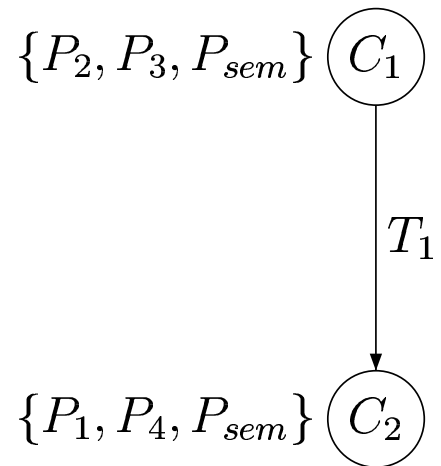
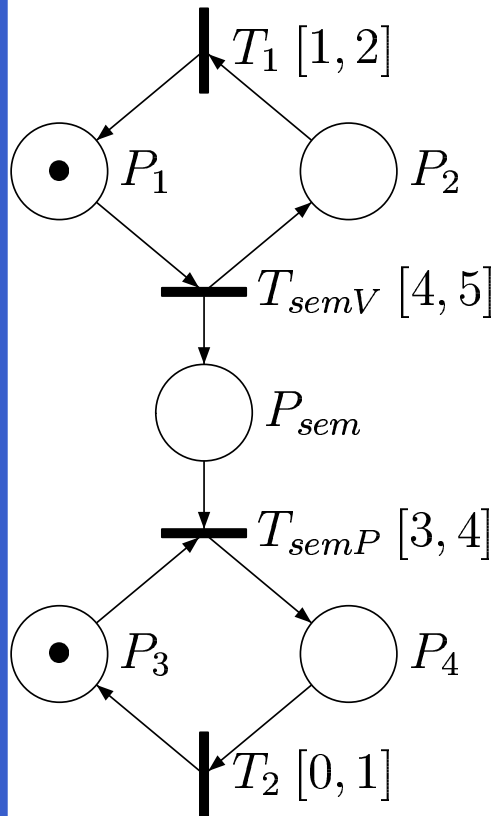
$$\exists (l_i, \delta, a, R, \rho, l_j) \text{ t.q. } \left\{ \begin{array}{l} \delta = (trans_i^{-1}(t) \geq \alpha(t)), \\ a = t, \\ R = trans_j^{-1}(\uparrow enabled(M_i, t)), \\ \forall x \in \chi_i, x' \in \chi_j \\ \text{t.q. } trans(x) = trans'(x') \\ \text{et } x' \notin R, \rho(x) = x'. \end{array} \right.$$

- $\forall C_i \in C^{ext}, Inv(l_i) = \bigwedge_{x \in \chi_i, t \in trans_i(x)} (x \leq \beta(t)).$

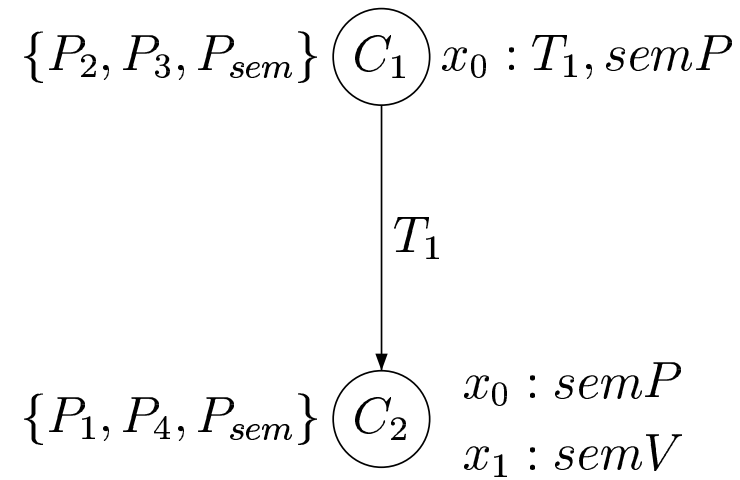
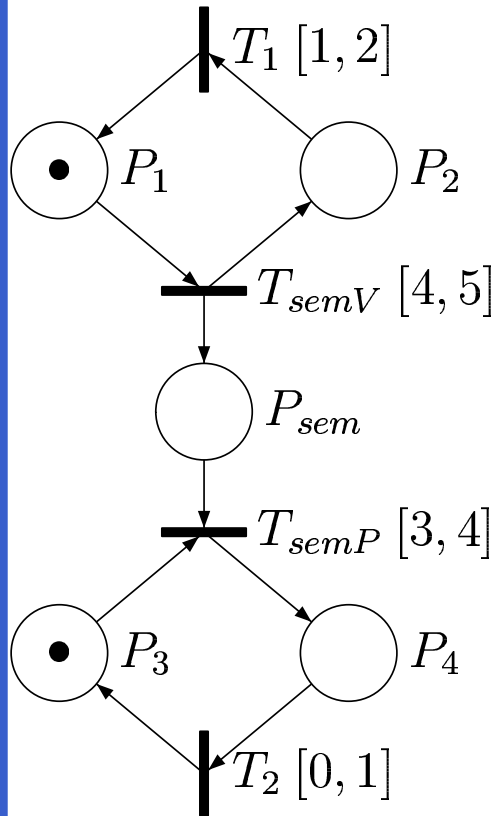
# AT des classes d'états - Exemple



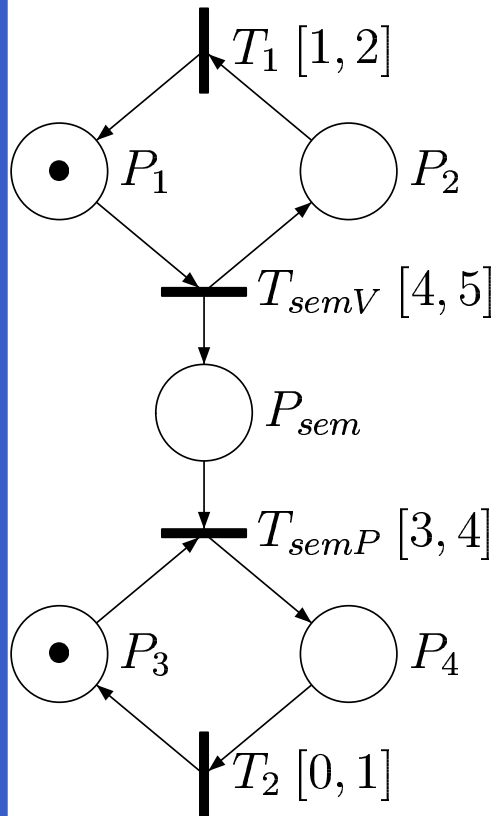
# AT des classes d'états - Exemple



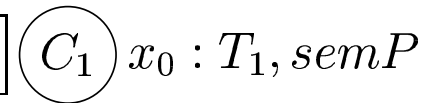
# AT des classes d'états - Exemple



# AT des classes d'états - Exemple

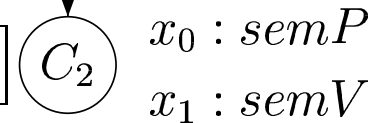


$$x_0 \leq \min\{\beta(T_1), \beta(semP)\}$$



$$T_1, x_0 \geq \alpha(T_1), x_1 := 0$$

$$x_0 \leq \beta(semP) \wedge x_1 \leq \beta(semV)$$



# AT des classes d'états - Résultats théoriques

- ▶ Le RdPT et son AT des classes d'états sont **temporellement bisimilaires**
- ▶ Le nombre d'horloge de l'AT est inférieur au nombre maximum de transitions sensibilisées simultanément et ne peut pas être réduit par l'algorithme classique de Daws et Yovine [DY96]
- ▶ Critères de **bornitude** (ou non-bornitude) identiques à ceux du graphe des classes [BD91]

# Romeo

Démonstration...

(<http://www.irccyn.ec-nantes.fr/irccyn/d/fr/equipes/TempsReel/logs/software-2-romeo>)

# AT des classes d'états - Nombre d'horloges

	BST98	SY96	Sav01	Sav01 + DY96	ROMEO
Exemple 1	NA	NA	12	8	6
Exemple 2	13	12	10	3	3
Exemple 3	14	14	14	2	2
Exemple 4	24	24	22	2	2
Exemple 5	31	29	23	3	2
Exemple 6	10	5	10	1	1
Exemple 7	NA	NA	20	11	7
Exemple 8	NA	NA	21	11	7
Exemple 9	NA	NA	15	3	3
Exemple 10	20	31	13	3	3
Exemple 11	12	20	9	4	4
Exemple 12	16	12	13	4	4
Exemple 13	20	16	17	4	4
Exemple 14	16	20	16	4	4

# AT des classes d'états - Taille

	Locations (TA)	Transitions (TA)	Nodes (Graph)	Transitions (Graph)
Example 1	123	258	355	661
Example 2	33	47	59	79
Example 3	16	16	16	16
Example 4	23	24	23	24
Example 5	48	69	159	206
Example 6	5	10	5	10
Example 7	1140	3990	14418	46079
Example 8	1277	4334	13557	41249
Example 9	58	135	252	548
Example 10	39	68	126	222
Example 11	50	123	138	330
Example 12	123	333	4256	8977
Example 13	407	1076	24401	50876
Example 14	998	3088	22016	60967

# Conclusion et perspectives

## Conclusion :

- ▶ Obtention d'un AT temporellement bisimilaire au RdPT de départ
- ▶ TCTL est décidable pour les RdPT bornés
- ▶ Un nombre "minimal" d'horloges
- ▶ Un nombre réduit de classes

## Perspectives :

- ▶ Diminution de la complexité du calcul de chaque classe
- ▶ Extension au Réseaux de Petri T-temporels étendus à l'ordonnancement
- ▶ Traduction structurelle inverse

# References

## References

- [BD91] Bernard Berthomieu and Michel Diaz. Modeling and verification of time dependent systems using time petri nets. *IEEE transactions on software engineering*, 17(3):259–273, 1991.
- [DY96] Conrado Daws and Sergio Yovine. Reducing the number of clock variables of timed automata. In *1996 IEEE Real-Time Systems Symposium (RTSS'96)*, pages 73–81, Washington, DC, USA, december 1996. IEEE Computer Society Press.
- [HNSY94] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.