Séquentialisation du comportement de réseaux de Petri temporisés

Jan Komenda, Sébastien Lahaye et Jean-Louis Boimond

Institut de Mathématiques, Academie Tchèque des Sciences,
Brno, République Tchèque
et
LISA, ISTIA
Angers, France

14 Novembre 2013
MSR 2013
INRIA Rennes, France

outline

Introduction

- Réseaux de Petri temporisés bornés et automates (max,+) : SED avec temporisation deterministe
- Synchronisation et partage des resources
- Comportements des automates (max,+): **séries formelles** avec coefficients dans le semi anneau: $\overline{\mathbb{R}}_{\max} = (R \cup \{-\infty\}, \max, +)$.
- Applications à l'évaluation des performances ou la commande supervisée automates (max,+)
- Méthodes connues pour l'obtention d'automates (max,+) représentant des RdPT : automates non déterministes et échec de la déterminisation
- Déterminisme du comportement : propriété importante pour certains résultats d'évaluation de performances et surtout pour la construction des côntrolleurs
- Nous allons présenter une méthode simple pour la construction d'automates déterministes représentant des RdPT bornés.

Définition (Réseaux de Petri temporisés bornés)

- $\mathcal{P}$ est un ensemble fini de places,
- $\mathcal{T}$ est un ensemble fini de transitions
- $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ est une relation entre les places et les transitions
- $M_0 : \mathcal{P} \to \mathbb{N}$ est le marquage initial
- $\tau = (\tau_a)_{a \in \mathcal{T}}$ est un vecteur représentant les temps associés aux transitions

Notation: pour $a \in \mathcal{T}$, ${}^\bullet a$ (resp., $a^\bullet$) est l'ensemble de ses places d'entrée (resp., de sortie) de a.
pour $p \in \mathcal{P}$, ${}^\bullet p$ **(resp.,** $p^\bullet$ est l'ensemble des transitions en amont (resp., en aval) de p.

Graphe d'atteignabilité

Le graphe d'atteignabilité d'un RdPT $\mathcal{G}$ est l'automate $\mathrm{Reach}(\mathcal{G}) = (\mathcal{M}, M_0, \mathcal{T}, t_r)$, où

- $\mathcal{M}$ est l'ensemble des marquages atteignables (M_0 marquage initial)
- $\mathcal{T}$ est l'ensemble des transitions (événements)
- $t_r : \mathcal{M} \times \mathcal{T} \to \mathcal{M}$ est la fonction partielle de transition: pour $M, M' \in \mathcal{M}$, $t \in \mathcal{T}$: $t_r(M,t) = M'$ ssi $M \xrightarrow{t} M'$.

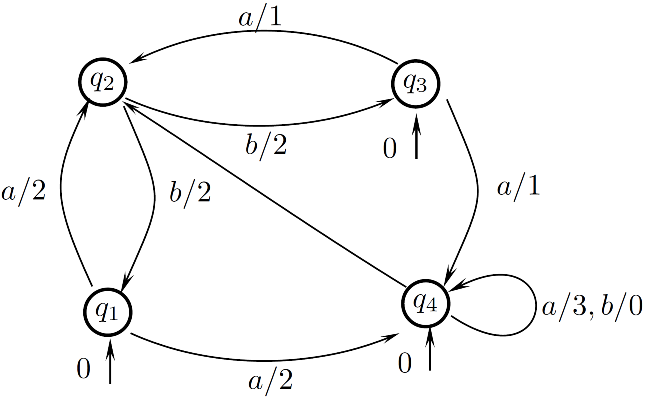
Un réseau de Petri est dit sauf (resp., *m*-borné) si pour tous les marquages accessibles, chaque place contient au plus un jeton (resp., au plus *m* jetons).

Automates (max,+)

- **Automates (max,+)** géneralise les automates logiques et les systèmes (max,+)-linéaires (e.g. graphes d'événements temporisé)
- Définis par $G = (Q, A, q_0, t)$ avec
  Q ensemble d'états (q_0 état initial),
  A ensemble d'événements,
  $t : Q \times A \times Q \to \mathbb{R}_{max}$ fonction de transition
  Interprétation: $t(q, a, q') \in \mathbb{R}_{max}$ correspond à la durée de $a-$transition de q vers q' et
  $t(q, a, q') = \varepsilon$ s'il n'y a pas de transition de q vers q' étiquetté par a.
- Automates (max,+) Déterministes: t est déterministe, i.e. $t : Q \times A \to Q \times \mathbb{R}_{max}$

Example d'automate (max,+)

Un automate (max,+)



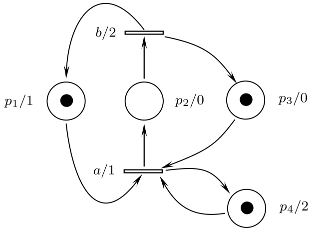
Modélisation des RdPT par automates (max,+)

Proposition. Soit $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$ un RdPT sauf. Automate (max,+) $G = (Q, A, \alpha, \mu)$ est derivé de $\mathcal{G}$ comme suit:

1. $Q = \mathcal{P}$,
2. $A = \mathcal{T}$,
3. $\forall q \in Q$,
$$\alpha_q = \begin{cases} e & \text{if } M_q = 1, \\ \varepsilon & \text{otherwise} \end{cases}$$
4. $\forall q, q' \in Q$, $\forall a \in A$,
$$[\mu(a)]_{qq'} = \begin{cases} \tau_a + \tau_q & \text{if } q \in {}^\bullet a \text{ and } q' \in a^\bullet, \\ e & \text{if } q = q' \text{ and } q \notin {}^\bullet a \cup a^\bullet, \\ \varepsilon & \text{otherwise} \end{cases}$$

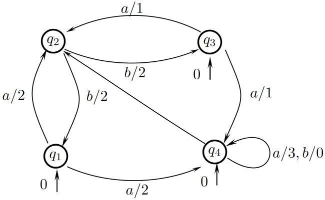
Alors,
$$x_G(w) = x_{\mathcal{G}}(w)\ ,$$
pour tout $w \in \mathcal{T}^*$ a possible séquence de $\mathcal{G}$ (i.e., $w \in L(\mathcal{G})$).

Example de modélisation.



RdPT $\mathcal{G}$ de JobShop.

Automate (max,+) G correspondant non déterministe.



Notions de base des réseaux de Petri temporisés

Sémantique opérationnelle en terme de Systèmes de Transitions Temporisés (STT)

- **L'état étendu** $(M, \mathbf{c})$ où $\mathbf{c} = (c_t)_{t \in \mathcal{T}}$ est **vecteur des horloges.**
- L'état sémantique initial $(M_0, \mathbf{0})$, où $\mathbf{0} = (0 \ldots 0)$
- transition $t \in \mathcal{T}$ est **sensibilisée** par M (noté par $M \xrightarrow{t}$) si $\forall p \in {}^\bullet t : M(p) \geq 1$:

  Notation $En(M) = \{t \in \mathcal{T} : \ M \xrightarrow{t}\}$.
- Marquage intermédiaire :

  $$M''(p) = \begin{cases} M(p) & \text{si } t \notin p^\bullet, \\ M(p) - 1 & \text{sinon, } \textit{i.e.}, \ t \in p^\bullet. \end{cases}$$
- Transition **nouvellement sensibilisée:** $t \in NewEn(M, \bar{t}, M')$ si $M' \xrightarrow{t'}$ et soit $t' = \bar{t}$ ou bien $M'' \not\xrightarrow{t'}$.
- Notation:

  $$Reset(\mathbf{c}, S)_t = \begin{cases} c_t & \text{si } t \notin S \\ 0 & \text{si } t \in S \end{cases}$$

Sémantique opérationnelle des réseaux de Petri temporisés

- Politique de résolution des conflits: la course entre les transitions sensibilisées
- Politique de course diffère de la politique de préselection (tous les cas logiques possibles) !
- Formellement, STS suivant:
  $(M_0, \mathbf{0})$ et
  $(M, \mathbf{c}) \xrightarrow{a} (M', \mathbf{c}')$ si :
  il existe une transition $M \xrightarrow{a} M'$ dans le réseau logique ;
- $c_a = \tau_a$;
- $c' = Reset(\mathbf{c}, NewEn(M', t, M))$.

Automate (max,+) directe (sémantique)

On combine la transition discrète et continue (écoulement du temps) consécutive
au sein d'une seule transition à multiplicité!

On définit $(M, \mathbf{c}) \xrightarrow{a} (M', \mathbf{c} + d) = (M', \mathbf{c}')$ avec

$$d = \min_{a \in En(M)} (\tau_a - c_a)$$

(exactement calculable) et

$$\mathbf{c}' = Reset(\mathbf{c} + d, NewEn(M', t, M))$$

comme une transition d'automate (max,+)

$$(M, \mathbf{c}) \xrightarrow{a/d} (M', \mathbf{c}').$$

Présentation formelle de la procédure

Procédure de déterminisation du comportement d'un réseau de Petri temporisé borné $\mathcal{G} = (\mathcal{P}, \mathcal{T}, M_0, \mathcal{F}, \tau)$.

Idée derrière: dépliage du graphe de marquage en utilisant les vecteurs d'horloges

1. Construire le graphe de marquages de $\mathcal{G}$, noté $reach(\mathcal{G}) = (\mathcal{M}, A, M_0, t)$.
2. Construire l'automate (max,+) correspondant $G_d = (Q_d, A, q_{0,d}, \delta_d)$:

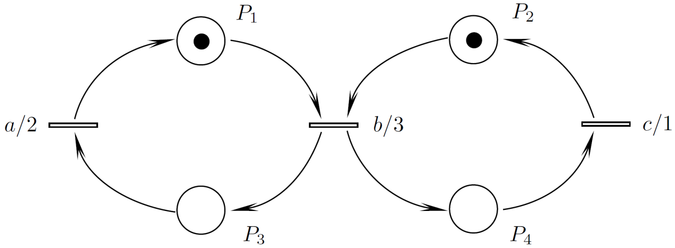
   Initialisation : $A = \mathcal{T}$, $q_0 = (M_0, (0, \ldots, 0))$,
   $Q_d = \{(M_0, (0, \ldots, 0))\}$,
   $\texttt{ProcédureRécursive}(M_0, (0, \ldots, 0))$.

Procedure

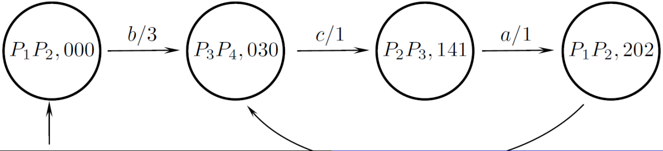
`ProcédureRécursive`$(M, \mathbf{c})$
Pour tout $a \in \mathcal{T}$ tel que $M \xrightarrow{a} M'$:
Calculer les nouvelles valeurs d'horloges c' pour la transition $(M, \mathbf{c}) \xrightarrow{a} (M', \mathbf{c}')$.
S'il existe $(\hat{M}, \hat{\mathbf{c}}) \in Q_d$ tel que $\hat{M} = M'$ et $\hat{\mathbf{c}} = \mathbf{c}'$
posons alors
$\delta_d((M, \mathbf{c}), a, (\hat{M}, \hat{\mathbf{c}})) = (\tau_a - c_a)^+$
Sinon
"Créer un nouvel état" $Q_d := Q_d \cup (M', \mathbf{c}')$ et posons
$\delta_d((M, \mathbf{c}), a, (M', \mathbf{c}')) = (\tau_a - c_a)^+$
`ProcédureRécursive`$(M', \mathbf{c}')$
Fin Pour tout
Fin `ProcédureRécursive`

Example 1.

Réseau de Petri temporisé $\mathcal{G}$.



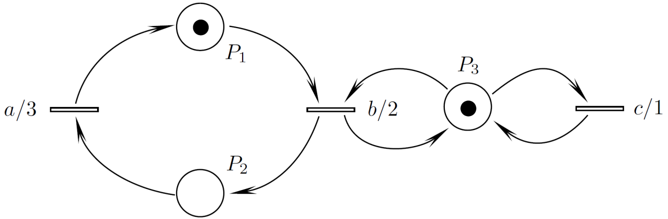
Automate (max,+) déterministe G correspondant à $\mathcal{G}$.

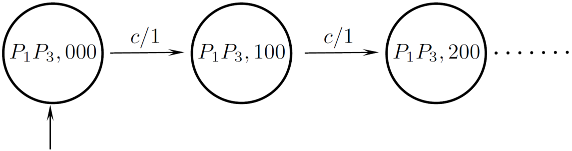


Problème: la procédure ne termine pas toujours!

Example 2. Problème typique : nombre infini d'états sémantiques (en dépit de temporisations discrètes)

Réseau de Petri pour lequel la procédure ne termine pas.





Condition suffisante pour la séquentialisation

Définition. (Vivacité forte) RdPT borné $\mathcal{G}$, est dit fortement vivant si $(\forall t \in \mathcal{T},\ \forall w \in L(\mathcal{G}),\ \forall v \preceq w,\ \exists N \in \mathbb{N} : (\mid v \mid \geq N \quad \Rightarrow \quad t \in v))$.

Proposition. Soit $\mathcal{G}$ un réseau de Petri temporisé qui satisfait la vivacité forte. Son comportement est alors séquentialisable, *i.e.,* il existe un automate (max,+) déterministe fini ayant la même série formelle.

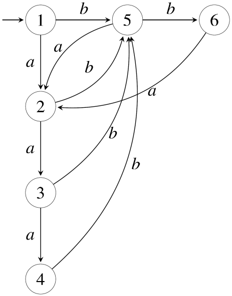
Corollaire. pour tout graphe d'événements temporisé $\mathcal{G}$ borné, il existe un automate (max,+) déterministe fini ayant la même série formelle que $\mathcal{G}$.

Verification de vivacité forte

Proposition. La vivacité forte d'un réseau de Petri temporisé borné $\mathcal{G}$ équivaut à ce que chaque circuit de son graphe de marquages contienne toutes les transitions du réseau.
Remarque. Les circuits du graphe de marquages : T-semiflots, i.e. solutions $S \in \mathbb{N}^{\mathcal{T}}$ de $W.S = \mathbf{0}$, où
W est la matrice d'incidence
Les solutions sont des vecteurs d'entiers naturels S, dont les composantes correspondent au nombre d'occurrences de chaque transition dans le circuit.
On vérifie si toutes les composantes de S sont positive.

Séquentialisation par restriction de langages

Restriction de langages est l'arbitrage partiel des conflits!



Automate restreignant le langage $(a \oplus b)^*$.

On considère le produit synchrone du graphe de marquage original avec les automates comme celui sur la Figure

Conclusion

- Automates (max,+) correspondant aux réseaux temporisés (modélisation directe) sont non déterministes
- Automates (max,+) issus de la sémantique sont déterministe mais infinis
- Condition suffisante pour qu'il soient finis
- Rafinement du langage pour les rendre finis
- Une condition structurelle (autre travail en parallèle)
- Applications (commande, évaluation de performances,...)