La Méthode des Classes d'États pour l'Analyse des Réseaux Temporels

Mise en Œuvre, Extension à la multi-sensibilisation

Bernard Berthomieu

LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse Cedex
Bernard.Berthomieu@laas.fr

RÉSUMÉ. *Cet article propose une extension de la technique d'analyse des réseaux de Petri Temporels dite "des classes d'états", et en décrit une mise en œuvre. Les réseaux Temporels étendent les réseaux de Petri en associant un intervalle temporel à chaque transition. Ils peuvent être utilisés pour la spécification et vérification de systèmes temps-réel ou de protocoles faisant intervenir des contraintes temporelles. La technique des classes d'états produit pour une large classe de réseaux Temporels une représentation finie de leur comportement, elle permet pour ces réseaux une analyse d'accessibilité semblable à celle permise pour les réseaux de Petri par la technique du graphe des marquages. La méthode des classes d'états est ici étendue à l'interprétation des transitions multi-sensibilisées. Une mise en œuvre de la construction du graphe des classes est décrite en détails.*

ABSTRACT. *This paper proposes an extension of the so called "state-classes" analysis method for Time Petri nets, and describes an implementation of the method. Time petri nets are obtained from Petri nets by associating a temporal interval with each transition. They can be used as a specification and verification tool for realtime systems or time dependant protocols. The state-classes technique allows to build, for a large class of Time Petri nets, a finite representation of their behavior. It allows for these nets a reachability analysis similar to that allowed for Petri nets by the marking graph method. The state-classes method is here extended by an interpretation of multi-enabledness of transitions. An implementation of the analysis technique is described precisely.*

MOTS-CLÉS : *Réseaux de Petri, Réseaux Temporels, Classes d'états, Multi-sensibilisation, Mise en œuvre.*

KEYWORDS: *Petri nets, Time Petri nets, State classes, Geometric regions, Multi-enabledness, Implementation.*

1. Introduction

Parmi les techniques proposées pour spécifier et vérifier des systèmes dans lesquels le temps apparaît comme paramètre, deux ont été développées à partir des réseaux de Petri : il s'agit d'une part des *Réseaux Temporisés* (ou *Timed Petri Nets*) [RAM 74], et d'autre part des *Réseaux Temporels* (ou *Time Petri Nets*) [MER 74].

Les réseaux Temporisés sont obtenus à partir des réseaux de Petri en associant une durée de tir à chaque transition. Les transitions sont tirées dès qu'elles sont sensibilisées. Ces réseaux sont essentiellement utilisés pour l'analyse de performances.

Les réseaux Temporels sont obtenus en associant deux dates min et max à chaque transition. Supposons que t soit devenue sensibilisée à la date θ, alors t ne peut être tirée avant la date $\theta + min$ et doit l'être au plus tard à la date $\theta + max$, sauf si le tir d'une autre transition a désensibilisé t avant que celle-ci ne soit tirée. Le tir des transitions est de durée nulle. Les réseaux Temporels expriment nativement des spécifications "en délais". En explicitant débuts et fins d'actions, ils peuvent aussi exprimer des spécifications "en durées". Leur domaine d'application est donc large.

Nous présentons dans cet article une extension et une mise en œuvre de la méthode d'analyse par énumération pour les réseaux Temporels développée dans [BER 82] [MEN 82] [BER 83] [BER 91]. Cette méthode, dite *des classes d'états*, permet pour une large classe de réseaux Temporels une analyse d'accessibilité semblable à la méthode du graphe des marquages utilisée pour l'analyse d'accessibilité des réseaux de Petri. L'extension concerne l'interprétation des transitions multi-sensibilisées, selon une approche suggérée dans [BER 82] mais non encore formellement exposée. Une mise en oeuvre est proposée, avec une description détaillée de l'algorithmique d'obtention des classes d'états, déterminante pour une implantation efficace de la méthode.

Les concepts de base des réseaux Temporels sont rappelés Section 2. La Section 3 résume la technique d'analyse par les classes d'états. Le traitement proposé de la multi-sensibilisation est exposé Section 4. La mise en œuvre de la méthode est détaillée Section 5, par une description de l'outil logiciel *Tina* développé par l'auteur. En conclusion, la technique d'analyse est brièvement comparée à quelques alternatives.

2. Les Réseaux de Petri Temporels

2.1. *Réseaux Temporels*

Définition 1 *Un réseau de Petri Temporel est un tuple* $\langle P, T, \mathbf{Pre}, \mathbf{Post}, M_0, \mathbf{IS}\rangle$, *dans lequel* $\langle P, T, \mathbf{Pre}, \mathbf{Post}, M_0\rangle$ *est un réseau de Petri, et* $\mathbf{IS} : T \rightarrow \mathbf{Q}^+ \times (\mathbf{Q}^+ \cup \{\infty\})$ *est la fonction Intervalle Statique.*

L'application $\mathbf{IS}$ associe à chaque transition t du réseau un intervalle à bornes rationnelles $\mathbf{IS}(t) = [min, max]$ avec $0 \leq min \leq max$ et max pouvant être ∞. La figure 1 représente un réseau Temporel.

ϕ étant un intervalle de $\mathbf{Q}^+ \times (\mathbf{Q}^+ \cup \{\infty\})$, nous noterons $Min(\phi)$ sa borne inférieure, et $Max(\phi)$ sa borne supérieure si celle-ci est finie, ou ∞ sinon.

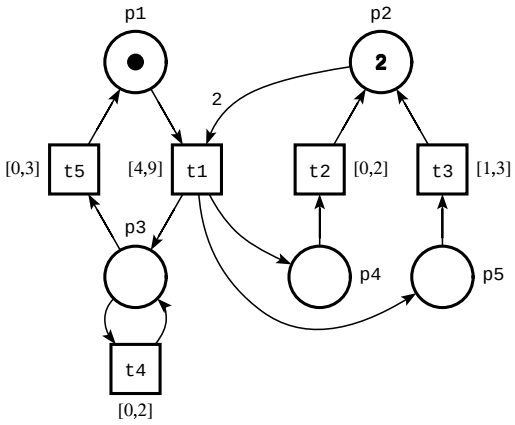


FIG. 1: Un réseau Temporel

2.2. *Comportement, états et relation d'accessibilité*

Définition 2 *Un État d'un réseau Temporel est un couple $E = (M, I)$ dans lequel M est un marquage et l'application I associe un intervalle temporel à chaque transition.*

L'état initial est constitué du marquage initial M_0 et de l'application I_0 qui associe à chaque transition sensibilisée son intervalle statique :

$E_0 = (M_0, I_0)$, avec $I_0(k) = \ si\ M_0 \geq \mathbf{Pre}(k)\ alors\ \mathbf{IS}(k)\ sinon\ \emptyset$

Toute transition sensibilisée doit être tirée dans l'intervalle de temps qui lui est associé. Cet intervalle est relatif à la date de sensibilisation de la transition. Franchir t, à une date relative θ, depuis un état $E = (M, I)$, est donc permis si et seulement si :

$M \geq \mathbf{Pre}(t) \wedge \theta \in I(t) \wedge \forall k \neq t.\ M \geq \mathbf{Pre}(k) \Rightarrow \theta \leq Max(I(k))$.

L'état $E' = (M', I')$ atteint depuis E par le tir de t à θ est alors déterminé par :

1) $M' = M - \mathbf{Pre}(t) + \mathbf{Post}(t)$ (comme dans les réseaux de Petri)

2) Pour chaque transition k :

- Si k est non sensibilisée par M', alors $I'(k) = \emptyset$
- Si k est distincte de t, sensibilisée par M, et non en conflit avec t, alors

$$I'(k) = [\max(0, Min(I(k)) - \theta), Max(I(k)) - \theta]$$

- Sinon, $I'(k) = \mathbf{IS}(k)$.

La règle de tir ci-dessus définit une relation d'accessibilité sur l'ensemble des états d'un réseau Temporel. Les *séquences de tirs* sont définies comme pour les réseaux de Petri, un *échéancier de tir* associe une séquence de tir s à une séquence u de dates. Un échéancier (s, u) est dit *réalisable* depuis un état E si les transitions de la séquence s sont successivement tirables depuis l'état E, aux dates relatives de tir qui leur correspondent dans la séquence u. Le fonctionnement d'un réseau Temporel peut être caractérisé par l'ensemble des états accessibles depuis son état initial ou, de façon duale, par l'ensemble des échéanciers réalisables depuis son état initial.

Notons que le concept d'état présenté associe exactement un intervalle à chaque transition sensibilisée, que celle-ci soit ou non multi-sensibilisée (*t* est *multi-sensibilisée* par M s'il existe un entier $k > 1$ tel que $M \geq k.\mathbf{Pre}(t)$). Cette interprétation de la sensibilisation sera qualifiée de *standard*, une alternative sera proposée Section 4.

Construire le graphe d'accessibilité des états d'un réseau Temporel est en général impossible : les transitions pouvant être tirées à tout instant dans leur intervalle de tir, les états admettent en général une infinité de successeurs. Les *Classes d'États* introduites en Section 3 ont pour but de fournir une représentation finie de ce graphe.

2.3. *Illustration*

Il y a plusieurs façons de représenter l'information temporelle des états d'un réseau Temporel. Pour éclairer la suite de l'exposé, un état sera représenté par une paire (M,D), dans laquelle D est un ensemble de vecteurs de dates de tir, appelé domaine de tir. Les vecteurs de D ont une composante pour chaque transition sensibilisée par M ; la i-ème projection de D est l'intervalle de tir $I(t_i)$ associé à la i-ème transition sensibilisée. Ces domaines de tir peuvent être exprimés comme l'ensemble des solutions de systèmes d'inéquations linéaires avec une variable associée à chaque transition sensibilisée (notées comme ces transitions).

L'état initial $E_0 = (M_0, D_0)$ du réseau Figure 1 est ainsi représenté par :

M_0 : $p_1, p_2(2)$
D_0 : Ensemble des solutions en t_1 du système $4 \leq t_1 \leq 9$

Le tir de t_1 depuis E_0, à une date relative $\theta_1 \in [4, 9]$, mène en $E_1 = (M_1, D_1)$:

M_1 : p_3, p_4, p_5
D_1 : Ensemble des solutions en (t_2, t_3, t_4, t_5) de
$0 \leq t_2 \leq 2$
$1 \leq t_3 \leq 3$
$0 \leq t_4 \leq 2$
$0 \leq t_5 \leq 3$

Le tir de t_2 depuis E_1, à une date relative $\theta_2 \in [0, 2]$, mène en $E_2 = (M_2, D_2)$:

M_2 : p_2, p_3, p_5
D_2 : Ensemble des solutions en (t_3, t_4, t_5) de

$\max(0, 1 - \theta_2) \leq t_3 \leq 3 - \theta_2$
$0 \leq t_4 \leq 2 - \theta_2$
$0 \leq t_5 \leq 3 - \theta_2$

Le paramètre θ_2 pouvant prendre toute valeur réelle dans l'intervalle $[0, 2]$, l'état E_1 admet une infinité de successeurs par le tir de t_2. Un exemple d'échéancier réalisable depuis l'état initial est $(t_1.t_2, 5.0)$.

3. Analyse d'Accessibilité, méthode des classes d'états

3.1. *Classes d'états*

Comme déjà mentionné, l'ensemble des états d'un réseau Temporel est en général infini, et ceci pour deux raisons : d'une part un état peut admettre une infinité (non dénombrable) de successeurs, et, d'autre part, un réseau peut admettre des échéanciers passant par des états tous différents. Le deuxième problème sera discuté Section 3.4. Pour résoudre le premier, une solution est bien sûr de regrouper certains états.

Une possibilité est de regrouper tous les états obtenus depuis l'état initial par le tir d'une même séquence de tir. Tous ces états ont même marquage, et leurs domaines de tir ne varient que par un décalage de certaines composantes et une troncature (leur réunion est un ensemble convexe). En termes de comportements, ce groupement préserve les traces maximales, et donc les propriété de sûreté (safety), incluant les blocages.

Considérons donc l'ensemble de tous les états que l'on peut atteindre depuis l'état initial par le tir d'échéanciers ayant pour support la même séquence de tir s. Cet ensemble définit la *Classe d'États* associée à la séquence de tir s. Les classes d'états peuvent être représentées par un marquage et un domaine de tir, comme l'étaient les états au Paragraphe 2.3. Le marquage est celui des états agglomérés dans la classe, le domaine de tir de la classe est la réunion des domaines de tir des états constituant la classe.

Définition 3 *Une classe d'états est un couple $C = (M, D)$ dans lequel M est un marquage, et $D = \{\underline{d} \mid A.\underline{d} \leq b\}$ est un domaine de tir. Les vecteurs $\underline{d}$ ont une composante pour chaque transition sensibilisée par M.*

Si $\underline{d} \in D$, nous noterons $\underline{d}_t$ la composante de $\underline{d}$ relative à la transition t. On définit ensuite une relation d'accessibilité entre classes d'états, comme suit.

3.2. *Transitions entre Classes d'états*

La classe initiale coïncide avec l'état initial du réseau, il a la forme requise s'il est présenté comme en Section 2.3.

Une transition t est tirable depuis une classe $C = (M, D = \{\underline{d} \mid A.\underline{d} \leq b\})$ si :

i) $M \geq \mathbf{Pre}(t)$ (t est sensibilisée par M)

ii) le système $A.\underline{d} \leq b$ augmenté des contraintes $\forall k \neq t.\ \underline{d}_t \leq \underline{d}_k$ est consistant (t est tirée dans son intervalle de tir et peut l'être avant les autres transitions sensibilisées)

Le calcul de la classe successeur $C' = (M', D')$ est alors effectué comme suit :

1) $M' = M - \mathbf{Pre}(t) + \mathbf{Post}(t)$ (comme dans les réseaux de Petri)

2) Le domaine D' est déterminé en quatre étapes :

a) Les conditions (ii) de franchissement de t sont ajoutées au système $A.\underline{d} \leq b$

b) Les variables associées aux transitions en conflit avec t sont éliminées

c) Chaque variable $\underline{d}_k$, $k \neq t$, est remplacée par $\underline{d}_t + \underline{d}_k$; $\underline{d}_t$ est ensuite éliminée

d) Pour chaque transition k nouvellement sensibilisée, on ajoute les contraintes :

$$Min(\mathbf{IS}(k)) \leq \underline{d}_k \leq Max(\mathbf{IS}(k))$$

L'ensemble des solutions du système déterminé à l'étape (c) peut être vu comme le domaine de tir des transitions distinctes de t qui sont restées sensibilisées pendant le tir de t, exprimé avec pour nouvelle origine du temps la date à laquelle la transition t a été tirée. Les éliminations effectuées dans les étapes (b) et (c) préservent les contraintes temporelles induites sur les variables restantes. Pour cette opération, on peut utiliser la méthode d'élimination classique de Fourier-Motzkin [DAN 63].

On laissera au lecteur le soin de vérifier que les domaines de tir des classes ainsi construites incluent exactement les domaines des états qu'elles agglomèrent. Deux classes sont égales si et seulement si leurs marquages et domaines respectifs sont égaux. La comparaison de domaines sera discutée en détail dans la Section 5. La relation d'accessibilité entre classes d'états définie ci-dessus permet de construire un graphe des classes : il contient la classe initiale, et il y a un arc étiqueté t d'origine C et d'extrémité C' si t est tirable depuis la classe C et son tir depuis C conduit en C'.

3.3. *Illustration*

A titre d'illustration, construisons quelques classes du réseau Temporel représenté Figure 1. La classe initiale C_0 est définie comme l'état initial E_0 (voir Section 2.3). Le tir de t_1 depuis C_0 conduit à une classe C_1 identique à l'état E_1 (cf. Section 2.3). Le tir de t_2 depuis C_1 conduit à la classe $C_2 = (M_2, D_2)$, avec $M_2 = (p_2, p_3, p_5)$, et D_2 calculé en 4 étapes, selon la règle ci-dessus :

Étape (a) : $D_2(a)$ est obtenu en ajoutant à D_1 les conditions de tirabilité de t_2 :

$$\begin{array}{ll} 0 \leq t_2 \leq 2 & t_2 \leq t_3 \\ 1 \leq t_3 \leq 3 & t_2 \leq t_4 \\ 0 \leq t_4 \leq 2 & t_2 \leq t_5 \\ 0 \leq t_5 \leq 3 & \end{array}$$

Étape (b) : Aucune transition n'étant en conflit avec t_2, on a $D_2(b) = D_2(a)$.

Étape (c) : Le changement d'origine produit le système suivant :

$$\begin{array}{lll} 0 \leq t_2 \leq 2 & 1 \leq t_2 + t_3 \leq 3 & t_2 \leq t_2 + t_3 \\ & 0 \leq t_2 + t_4 \leq 2 & t_2 \leq t_2 + t_4 \\ & 0 \leq t_2 + t_5 \leq 3 & t_2 \leq t_2 + t_5 \end{array}$$

Depuis lequel $D_2(c)$ est obtenu par élimination de t_2 :

$$\begin{array}{ll} 0 \leq t_3 \leq 3 & t_4 - t_3 \leq 1 \\ 0 \leq t_4 \leq 2 & t_5 - t_3 \leq 2 \\ 0 \leq t_5 \leq 3 & \end{array}$$

Étape (d) : Aucune transition n'étant nouvellement sensibilisée, on a $D_2 = D_2(c)$.

Le graphe des classes complet du réseau Figure 1 est donné en Annexe, tel que produit par l'outil *Tina* décrit en Section 5.

3.4. *Caractère Borné*

Toute classe a un nombre fini de successeurs (au plus un par transition sensibilisée). Il reste à examiner les conditions sous lesquelles l'ensemble des classes est fini.

Rappelons qu'un réseau de Petri est *Borné* si le marquage de toute place admet une borne supérieure. La propriété Borné est indécidable pour les réseaux Temporels (voir par exemple [MEN 82]), mais l'ensemble des domaines de tir d'un réseau Temporel est toujours fini [BER 82]. Le graphe des classes d'un réseau Temporel est donc fini si et seulement si ce réseau est borné.

Ainsi, toute condition suffisante pour la propriété Borné fournira une condition suffisante pour la propriété de finitude du graphe des classes. Le théorème 4 [BER 82] énonce quelques unes de ces conditions.

Théorème 4 *Un réseau Temporel est borné s'il n'admet pas de paire de classes d'états $C = (M, D)$ et $C' = (M', D')$ telles que :*

i) C' *est accessible depuis* C

ii) $M' \gtrsim M$

iii) $D' = D$

iv) $\forall p. M'(p) > M(p) \Rightarrow M'(p) \geq \max_{(t \in T)}\{\mathbf{Pre}(p,t)\}$

Les propriétés (i) à (iv) sont nécessaires pour qu'un réseau soit non borné, mais pas suffisantes. Ce théorème permet, par exemple, de démontrer que les réseaux Figures 1, 2(a) et 2(b) sont bornés, mais il ne permet pas de démontrer que le réseau 2(c) est borné bien que celui-ci n'admette que 48 classes d'états.

L'omission de le clause (iv) ou de la clause (iii) fournit des conditions suffisantes plus fortes. (iv) omise, on ne peut plus montrer que le réseau 2(b) est borné. Omettant de plus (iii), le réseau 2(a) ne peut être montré borné ; la condition suffisante obtenue est alors celle qui permet de décider si un réseau de Petri ordinaire est borné [KAR 69].

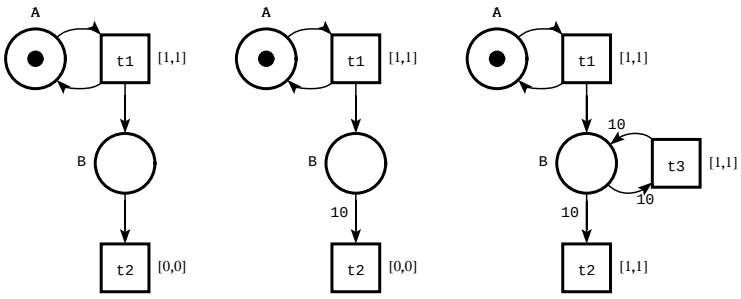


FIG. 2: Trois réseaux Temporels bornés

Pour les réseaux que l'on ne peut montrer bornés à l'aide du Théorème 4, il reste possible de procéder à une énumération contrainte, par exemple en bornant arbitrairement le nombre de classes à énumérer ou le marquage de chaque place. L'analyse structurelle fournit aussi des conditions suffisantes pour le caractère borné.

4. Interprétations de la multi-sensibilisation

4.1. *Sensibilisations multiples*

Une transition t est *multi-sensibilisée* par un marquage M s'il existe un entier $k > 1$ tel que $M \geq k.\mathbf{Pre}(t)$. Dans l'interprétation standard de la sensibilisation, utilisée dans les Sections 2 et 3, chaque transition sensibilisée est associée à une et une seule variable temporelle du domaine de tir, qu'elle soit ou non multi-sensibilisée. Nous explorons ici d'autres interprétations, qualifiées d'*étendues*, dans lesquelles les transitions multi-sensibilisées sont associées à plusieurs variables temporelles.

Le réseau Figure 3 ci-dessous sera utilisé pour illustrer ces interprétations.

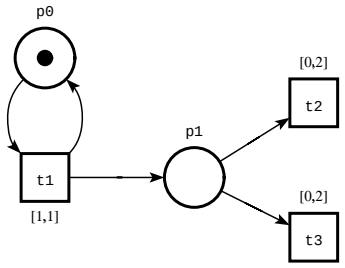


FIG. 3: Illustration de la multi-sensibilisation

Le tir de t_1 depuis la classe initiale C_0 de ce réseau conduit dans la classe C_1 :

$M_1 : p_0, p_1$
D_1 : Ensemble des solutions en (t_1, t_2, t_3) de
$1 \leq t_1 \leq 1$
$0 \leq t_2 \leq 2$
$0 \leq t_3 \leq 2$

Par la règle standard, le tir de t_1 depuis C_1 conduit dans la classe C_2^s suivante :

$M_2 : p_0, p_1(2)$
D_2^s : Ensemble des solutions en (t_1, t_2, t_3) de
$1 \leq t_1 \leq 1$
$0 \leq t_2 \leq 1$
$0 \leq t_3 \leq 1$

Les transitions t_2 et t_3 sont restées sensibilisées pendant le tir de t_1, ce qui explique le décalage de leurs intervalles vers l'origine dans le système D_2^s. Noter que si t_2 et t_3 n'étaient pas sensibilisées par M_1, alors elles seraient contraintes par leurs intervalles statiques car elles seraient alors considérées comme "nouvellement sensibilisées".

Les interprétations que nous envisageons dans ce paragraphe associent ces deux intervalles aux transitions t_2 et t_3 : elles seront considérées à la fois comme persistantes pour le tir de t_1, et comme nouvellement sensibilisées. Le tir de t_1 depuis la classe C_1 conduira ainsi à la classe C_2 suivante dans laquelle la transition t_2 (resp. t_3) est associée à deux variables temporelles notées t_2^0 et t_2^1 (resp t_3^0 et t_3^1)

$M_2 : p_0, p_1(2)$
D_2 : Ensemble des solutions en $(t_1, t_2^0, t_2^1, t_3^0, t_3^1)$ de
$1 \leq t_1 \leq 1$
$0 \leq t_2^0 \leq 1$
$0 \leq t_2^1 \leq 2$
$0 \leq t_3^0 \leq 1$
$0 \leq t_3^1 \leq 2$

4.2. *Règle de tir étendue*

Une transition t, sensibilisée par un marquage M, sera ici associée à k variables temporelles du domaine de tir, notées t^0, ...,t^{k-1}, où k est le plus grand entier positif tel que $M \geq k.\mathbf{Pre}(t)$. Afin de compléter l'interprétation, il est nécessaire de définir quelles instances de transitions sont considérées lorsqu'une transition est franchie ou lorsqu'elle est en conflit avec la transition tirée, et ce qu'il advient des autres.

Plusieurs choix peuvent être envisagés : les instances de sensibilisation peuvent être considérées comme indépendantes (stratégie *non déterministe*), ou encore peuvent être ordonnées selon leur âge (e.g. stratégie *première sensibilisée - première tirée*, désignée par *PSPT* dans ce qui suit). L'interprétation non déterministe est la plus générale, mais elle conduit à un plus grand nombre de classes, et le graphe des classes produit peut être non déterministe. D'autre part, ordonner les instances de sensibilisation

selon leur âge semble raisonnable lorsque ces instances représentent des occurrences d'événements : la stratégie *étendue PSPT* traduit alors le fait que les événements sont traités dans l'ordre de leur arrivée.

Notons qu'aucune de ces interprétations ne nécessite de distinguer les jetons. Dans tous les cas, la règle de tir du Paragraphe 3.2 est aisément adaptée. Pour l'interprétation *étendue PSPT*, les modifications nécessaires se résument à :

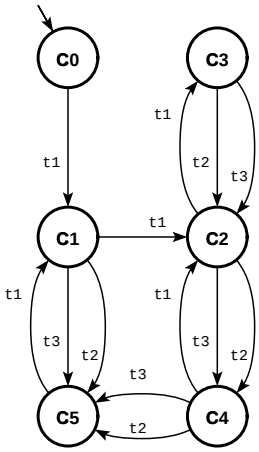
– Les variables de chaque système sont ordonnées selon l'ordre de leur introduction. Lors du tir d'une transition, on considère l'instance la plus ancienne.

– A l'étape (2b), on élimine les instances de transitions en conflit avec t, en commençant par les plus anciennes.

– A l'étape (2c), on translate toutes les instances des transitions qui sont restées sensibilisées pendant le tir de l'instance couramment tirée.

– A l'étape (2d), on introduit une nouvelle variable pour chaque instance de chaque transition nouvellement sensibilisée.

Pour l'exemple Figure 3, on obtient ainsi le graphe de classes représenté Figure 4.



C_0		C_1		C_2	
M_0	p_0	M_1	p_0, p_1	M_2	$p_0, p_1(2)$
D_0	$1 \leq t_1 \leq 1$	D_1	$1 \leq t_1 \leq 1$	D_2	$1 \leq t_1 \leq 1$
			$0 \leq t_2 \leq 2$		$0 \leq t_2^0 \leq 1$
			$0 \leq t_3 \leq 2$		$0 \leq t_2^1 \leq 2$
					$0 \leq t_3^0 \leq 1$
					$0 \leq t_3^1 \leq 2$
C_3		C_4		C_5	
M_3	$p_0, p_1(3)$	M_4	p_0, p_1	M_5	p_0
D_3	$1 \leq t_1 \leq 1$	D_4	$0 \leq t_1 \leq 1$	D_5	$0 \leq t_1 \leq 1$
	$0 \leq t_2^0 \leq 0$		$0 \leq t_2 \leq 2$		
	$0 \leq t_2^1 \leq 1$		$0 \leq t_3 \leq 2$		
	$0 \leq t_2^2 \leq 2$		$t_2 - t_1 \leq 1$		
	$0 \leq t_3^0 \leq 0$		$t_3 - t_1 \leq 1$		
	$0 \leq t_3^1 \leq 1$				
	$0 \leq t_3^2 \leq 2$				

FIG. 4: Graphe des classes du réseau Figure 3, stratégie de tir *PSPT*

4.3. *Propriété borné*

Le théorème 4 n'est plus applicable pour les interprétations étendues. La dimension des domaines de tir n'etant plus bornée par le nombre de transitions, la condition (iii) n'est plus nécessaire pour la propriété non-borné. Afin de garantir la terminaison, il suffit de la remplacer par la suivante :

> iii') $D'|D = D$, où $D'|D$ est obtenu en éliminant du système D' les variables n'apparaissant pas dans D.

En outre, chaque transition du réseau Temporel considéré doit posséder au moins un place d'entrée (sinon le nombre de variables du domaine initial serait infini).

Hélas, la condition suffisante pour la propriété borné ainsi exprimée est souvent trop forte, elle ne permet pas par exemple de décider que le réseau Figure 3 est borné. Si la propriété borné ne peut être démontrée ainsi, on aura recours à une énumération contrainte, en bornant arbitrairement le nombre de marquages, le marquage des places, ou encore le nombre d'instances de sensibilisation des transitions.

Enfin, il mérite d'être noté qu'un réseau Temporel peut être non borné avec l'interprétation standard mais borné avec l'interprétation étendue ! Ce serait par exemple le cas du réseau Figure 3 si t_2 et t_3 portaient l'intervalle $[2, 3]$ plutôt que $[0, 2]$.

5. Mise en œuvre

5.1. *L'outil Tina*

La technique d'analyse décrite Section 3 a été plusieurs fois mise en œuvre, dans des projets universitaires ou commerciaux. Une implantation, *Tina*, est proposée par l'auteur (*http ://www.laas.fr/~bernard/tina.html*). L'outil *Tina* est un descendant de l'outil de même nom décrit dans [ROU 86].

Depuis la description textuelle ou graphique d'un réseau Temporel, *Tina* construit son graphe des classes. Les tests d'arrêt disponibles incluent ceux exprimés par le Théorème 4. Les deux interprétations de la sensibilisation exposées dans cet article sont proposées (*standard* et *étendue PSPT*). Lorsque le graphe des classes est fini, *Tina* procède ensuite à une analyse de vivacité (aspects non abordés dans cet article).

Le résultat de l'invocation de *Tina* pour le réseau de la Figure 1 est montré en Annexe. Les paragraphes suivants décrivent les principaux algorithmes utilisés.

5.2. *Représentation des classes d'états*

Les classes associent un marquage et un domaine de tir. Le codage des marquages ne sera pas discuté ici ; les domaines sont représentés par deux vecteurs a et b de taille n, et une matrice c, de taille $n \times n$; où n est le nombre de transitions sensibilisées. Sans perte de généralité, on peut supposer que les composantes de a, b, et c sont des entiers (pouvant être infinis, pour b et c). Pour faciliter l'exposé, on supposera aussi que ces composantes sont indicées par les transitions sensibilisées.

La classe initiale est obtenue depuis le marquage initial et l'application intervalle statique. Soit E l'ensemble des transitions sensibilisées par M_0, alors $C_0 = (M_0, D_0)$, avec $D_0 = (a, b, c)$ initialisé comme suit :

$$pour\ tout\ i \in E\ \{a_i := Min(\mathbf{IS}(i))\ ; b_i := Max(\mathbf{IS}(i))\}$$
$$pour\ tout\ j, k \in E\ \{c_{jk} :=\ si\ j \neq k\ alors\ b_j - a_k\ sinon\ 0\}$$

5.3. *Égalité des Classes d'Etats, Formes canoniques*

Deux classes sont égales si leurs marquages et domaines respectifs sont égaux. Pour comparer deux domaines, les systèmes qui les définissent seront mis sous forme canonique, puis ces formes canoniques comparées.

Tout domaine de tir peut être exprimé comme l'ensemble des solutions d'un système d'inéquations avec au plus deux variables par inéquation, de la forme suivante :

$a_i \leq t_i \leq b_i, \forall i \in E$
$t_j - t_k \leq c_{jk}, \forall j,k \in E, j \neq k$
avec $a_i \in \mathbf{Q}^+$, $b_i,\ c_{jk} \in \mathbf{Q}^+ \cup \{\infty\}$, et $E \subset T$

Ces systèmes admettent des formes canoniques de la même forme. Une technique pour les obtenir consiste à associer au système un *Graphe de Contraintes* [RAM 99]. La mise sous forme canonique se résume ensuite à un calcul de plus courts chemins entre toutes paires de sommets de ce graphe. Pour le calcul des plus courts chemins, on peut utiliser l'algorithme de Floyd-Warshall [COR 94], de complexité $O(n^3)$ en temps et $O(n^2)$ en espace (n est le nombre de sommets). Cet algorithme permet de plus de vérifier la consistance du système d'inéquations en s'assurant que le graphe de contraintes associé ne contient pas de cycle de poids négatif.

Adapté à notre représentation des systèmes d'inéquations, cette technique s'exprime comme suit. Soit V l'ensemble des variables du système à mettre sous forme canonique. Ajoutons temporairement à la représentation du système d'inéquations une nouvelle composante r (initialisée à 0). Cette composante r ne sert qu'à la détermination de consistance. On applique ensuite la procédure :

$pour\ tout\ k \in V$
  $pour\ tout\ i \in V$
    $pour\ tout\ j \in V$
      $r := \min(r, b_k - ak)$
      $a_j := \max(a_j, a_k - c_{kj})$
      $b_i := \min(b_i, b_k + c_{ik})$
      $c_{ij} := \min(c_{ij}, c_{ik} + c_{kj})$

Le système est consistant si et seulement si, en fin de traitement, $r \geq 0$ et, pour toute variable i, $c_{ii} \geq 0$. Si le système est consistant, alors il est sous forme canonique.

5.4. *Mise en oeuvre de la règle de tir*

Par souci de simplicité, nous ne détaillerons ici que la règle de tir relative à l'interprétation standard de la sensibilisation. L'implantation de la règle étendue n'en diffère que par la façon d'associer les variables des systèmes aux transitions du réseau.

Soit M le marquage courant, $D = (a, b, c)$ une copie du domaine courant, et E l'ensemble des transitions sensibilisées par M. Il s'agit de savoir si $k \in E$ peut être tirée depuis la classe courante, et, si c'est le cas, d'obtenir la classe résultante.

Le nouveau marquage est déterminé par $M' = M - \mathbf{Pre}(k) + \mathbf{Post}(k)$.

Le nouveau domaine D' est obtenu en quatre étapes, comme suit :

a) On ajoute les contraintes de tirabilité de k :

$pour\ tout\ i \neq k\ \{c_{ki} := \min(0, c_{ki})\}$

On remet ensuite le système sous forme canonique, et on vérifie qu'il est consistant. S'il ne l'est pas, alors la transition k ne peut être franchie depuis cette classe.

b) Le système étant sous forme canonique, l'élimination des variables associées aux transitions en conflit avec k se résume à la suppression des composantes correspondantes dans a, b et les lignes et colonnes de c.

c) Les effets combinés du décalage temporel et de la propagation des contraintes de la transition tirée s'expriment par

$pour\ tout\ i \neq k\ \{a_i := \max(0, a_i - b_k, -c_{ki})\ ; b_i := \min(b_i - a_k, c_{ik})\}$

On efface ensuite les composantes de a, b, et c correspondant à la transition tirée, et on met le système obtenu sous forme canonique.

d) Soit N l'ensemble des transitions nouvellement sensibilisées. On introduit $|N|$ nouvelles variables dans le système, contraintes comme suit

$pour\ tout\ i \in N\ \{a_i := Min(\mathbf{IS}(i))\ ; b_i := Max(\mathbf{IS}(i))\}$
$pour\ tout\ i, j \in N, k \notin N$
$\{c_{ij} := si\ i \neq j\ alors\ b_i – a_j\ sinon\ 0\ ; c_{ik} := b_i – a_k\ ; c_{kj} := b_k – a_j\}$

La règle de tir ainsi implantée nécessite deux mises sous forme canonique, une première pour déterminer la consistance, et une seconde pour obtenir la forme canonique finale. La première peut être évitée en utilisant les techniques incrémentielles de vérification de consistance [RAM 99], mais le bénéfice est ici faible. Une alternative est d'inverser l'ordre des étapes (b) et (c) en retardant la vérification de consistance de l'étape (a) jusqu'à la mise sous forme canonique finale (avant l'étape (d)). Cette variante obscurcit quelque peu l'énoncé de la règle de tir, mais donne de bons résultats.

5.5. *Construction du Graphe des Classes*

L'algorithme de construction du graphe des classes est basé sur l'algorithme de Tarjan pour le calcul des composantes fortement connexes maximales d'un graphe. Les différences sont que le graphe est ici construit à la demande par application de la règle de tir à la classe en tête de pile, et que l'on vérifie à la volée le caractère borné.

Les classes sont rangées dans un arbre binaire de recherche. Toute classe construite est comparée aux classes existantes par une relation d'ordre strict connexe *inf* choisie de façon à être compatible avec la relation exprimée par le Théorème 4 (c.a.d. telle que deux classes C et C' satisfaisant les conditions du théorème satisfassent C *inf* C').

La comparaison procède classiquement, excepté dans le cas où C *inf* C', C' étant la dernière classe construite, et C une classe rangée dans l'arbre. Dans ce cas, on vérifie à la volée que les conditions (i) à (iv) du théorème 4 ne sont pas simultanément

satisfaites. Si elles le sont, alors le réseau peut être non borné, et l'énumération est interrompue. Pour le test de la condition (i), notons que la pile courante de l'algorithme de Tarjan est exactement constituée des prédécesseurs de la classe construite. Cette propriété, associée à un codage de la pile dans les classes elle-mêmes, permet une implantation efficace du test du caractère Borné. L'autre usage de la pile est bien entendu le calcul des composantes fortement connexes, qui seront utilisées pour analyser la propriété de vivacité.

Enfin, *Tina* utilise deux représentations pour les classes d'états. Celle explicitée Section 5.2 n'est utilisée que pendant le tir des transitions. Une autre, plus compacte, est utilisée pour le stockage des classes et leur comparaison.

6. Conclusion

La méthode d'analyse exposée dans cet article permet pour les réseaux Temporels une analyse d'accessibilité semblable à celle permise pour les réseaux de Petri par la technique du graphe des marquages. Cette technique a été utilisée dans de nombreux travaux, universitaires ou industriels, et a été intégrée à plusieurs outils d'analyse de systèmes. Les limites intrinsèques de la méthode ne doivent toutefois pas être perdues de vue. Une première limite est qu'il ne peut être énoncé de condition nécessaire et suffisante pour la propriété borné pour les réseaux Temporels, une seconde est que le nombre de classes d'états d'un réseau Temporel peut être très grand.

Un possible frein au développement de cette méthode est son apparente complexité conceptuelle et calculatoire. Par un exposé détaillé de l'algorithmique requise, nous espérons avoir montré que cette complexité n'est qu'apparente. L'outil *Tina*, ainsi que d'autres, montre que l'analyse d'accessibilité des réseaux Temporels est praticable.

L'autre apport de cet article est le traitement de la multi-sensibilisation proposé Section 4. Les alternatives existantes sont principalement basées sur une datation des jetons, les instances de sensibilisation des transitions étant ordonnées selon l'âge des jetons qu'elles mobilisent [CER 99] [KHA 97]. La solution proposée Section 4 a l'avantage de la simplicité et de la généralité.

Concernant les applications de la méthode du graphe des classes à la vérification, certains auront noté que le groupement des états retenu préserve les propriétés de sûreté du graphe des états (séquences de tir, blocages), mais ne préserve pas notamment les propriétés de branchement et de vivacité. De récents travaux suggèrent des groupement d'états différents, les classes *atomiques* de [YON 98], par exemple, permettent la vérification de formules CTL sur le graphe des classes. Le concept de classe d'état utilisé dans ces travaux est différent du notre, mais le concept d'atomicité peut aisément être reformulé dans notre contexte ; ces développements seront exposés dans un futur article.

Enfin, en raison de leur similitude avec notre approche, nous ne saurions conclure sans mentionner les travaux sur les Automates Temporisés [ALU 94], introduit plus

récemment. Le modèle des Automates Temporisés ajoute à un automate classique un ensemble fini d'horloges, et annote les arcs de l'automate par des conditions et actions concernant les horloges. Bien que faisant aussi appel à des méthodes "géométriques", le traitement du temps dans les Automates Temporisés est significativement différent du traitement du temps dans les réseaux de Petri Temporels. Les Automates Temporisés gèrent notamment un nombre fini et constant d'horloges, alors que les réseaux de Petri Temporels créent dynamiquement des horloges au cours de l'évolution du marquage. Un rapport récent [HAA 00] présente des codages croisés entre ces deux techniques.

7. Bibliographie

[ALU 94] ALUR R., DILL D., « A theory of timed automata », *Theoretical Computer Science*, vol. 126, 1994, p. 183–235.

[BER 82] BERTHOMIEU B., MENASCHE M., « A state enumeration approach for analyzing Time Petri nets », *Applications and Theory of Petri Nets , Como, Italy*, 1982, p. 27–56.

[BER 83] BERTHOMIEU B., MENASCHE M., « An Enumerative Approach for Analyzing Time Petri Nets. », *IFIP Congress Series*, vol. 9, 1983, p. 41–46, North Holland.

[BER 91] BERTHOMIEU B., DIAZ M., « Modeling and Verification of Time Dependent Systems Using Time Petri Nets. », *IEEE Trans. on Soft. Eng.*, vol. 17, n° 3, 1991, p. 259–273.

[CER 99] CERONE A., MAGGIOLO-SCHETTINI A., « Time-based expressivity of time Petri nets for system specification. », *Theoretical Computer Science*, vol. 216, 1999, p. 1–53.

[COR 94] CORMEN T. H., LEISERSON C. E., RIVEST R. L., *Introduction à l'Algorithmique*, Dunod, Paris, 1994.

[DAN 63] DANTZIG G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ., 1963.

[HAA 00] HAAR S., KAISER L., SIMONOT-LION F., TOUSSAINT J., « On equivalence between Timed State Machines and Time Petri Nets. », rapport INRIA No 4049, nov. 2000.

[KAR 69] KARP R. M., MILLER R. E., « Parallel Program Schemata. », *Journal of Computer and System Sciences 3*, , n° 2, 1969, p. 147–195.

[KHA 97] KHANSA W., *Réseaux de Petri p-temporels : contribution à l'étude des systèmes à événements discrets*, Thèse de Doctorat de l'Université de Savoie, Annecy, France, 1997.

[MEN 82] MENASCHE M., *Analyse des Réseaux de Petri Temporisés et Applications aux Systèmes Distribués*, Thèse de Dr. Ingénieur de l'Université Paul Sabatier, Toulouse, 1982.

[MER 74] MERLIN P. M., *A Study of the Recoverability of Computing Systems.*, Irvine : Univ. California, PhD Thesis, 1974.

[RAM 74] RAMCHANDANI C., *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets.*, Cambridge, Mass. : MIT, Dept. Electrical Engineering, PhD Thesis, 1974.

[RAM 99] RAMALINGAM G., SONG J., JOSKOWICZ L., MILLER R. E., « Solving Systems of Difference Constraints Incrementally », *Algorithmica*, vol. 23, 1999, p. 261–275.

[ROU 86] ROUX J. L., BERTHOMIEU B., « Verification af a Local Area Network Protocol with TINA, a Software Package for Petri Nets. », *Applications and Theory of Petri Nets, Oxford, UK*, juillet 1986, p. 183–205.

[YON 98] YONEDA T., RYUBA H., « CTL Model Checking of Time Petri nets Using Geometric Regions », *IEEE Trans. on Information and Systems*, vol. E99-D, n° 3, 1998, p. 1-10.

Annexe : Invocation de Tina pour le réseau Figure 1

```
Tina version 2.0 -- November 2000 -- LAAS/CNRS

bounded, 12 classes, 29 transitions

STATE CLASSES:
class 0          class 1          class 2          class 3
p1 p2*2          p3 p4 p5         p2 p3 p5         p2*2 p3
4 <= t1 <= 9     0 <= t2 <= 2     0 <= t3 <= 3     0 <= t4 <= 1
                 1 <= t3 <= 3     0 <= t4 <= 2     0 <= t5 <= 2
                 0 <= t4 <= 2     0 <= t5 <= 3
                 0 <= t5 <= 3     t4 - t3 <= 1
                                  t5 - t3 <= 2

class 4          class 5          class 6          class 7
p2*2 p3          p2 p3 p5         p1 p2 p5         p2 p3 p4
0 <= t4 <= 2     0 <= t3 <= 3     0 <= t3 <= 3     0 <= t2 <= 1
0 <= t5 <= 3     0 <= t4 <= 2                      0 <= t4 <= 1
                 0 <= t5 <= 3                      0 <= t5 <= 2

class 8          class 9          class 10         class 11
p2 p3 p4         p1 p2 p4         p3 p4 p5         p1 p4 p5
0 <= t2 <= 1     0 <= t2 <= 1     0 <= t2 <= 2     0 <= t2 <= 2
0 <= t4 <= 2                      0 <= t3 <= 3     0 <= t3 <= 3
0 <= t3 <= 3                      0 <= t4 <= 2     t2 - t3 <= 1
                                  0 <= t5 <= 3
                                  t2 - t3 <= 1
REACHABILITY GRAPH:
0 -> t1 in [4,9]/1
1 -> t2 in [0,2]/2, t3 in [1,2]/7, t4 in [0,2]/10, t5 in [0,2]/11
2 -> t3 in [0,2]/3, t4 in [0,2]/5, t5 in [0,2]/6
3 -> t4 in [0,1]/4, t5 in [0,1]/0
4 -> t4 in [0,2]/4, t5 in [0,2]/0
5 -> t3 in [0,2]/4, t4 in [0,2]/5, t5 in [0,2]/6
6 -> t3 in [0,3]/0
7 -> t2 in [0,1]/3, t4 in [0,1]/8, t5 in [0,1]/9
8 -> t2 in [0,1]/4, t4 in [0,1]/8, t5 in [0,1]/9
9 -> t2 in [0,1]/0
10 -> t2 in [0,2]/5, t3 in [0,2]/8, t4 in [0,2]/10, t5 in [0,2]/11
11 -> t2 in [0,2]/6, t3 in [0,2]/9

possibly live

STRONG CONNECTED COMPONENTS:
0 : 0 1 2 3 4 5 6 7 8 9 10 11

SCC GRAPH:
0 -> t1/0, t2/0, t3/0, t4/0, t5/0
```