Sémantique concurrente symbolique des réseaux de Petri saufs et dépliages finis des réseaux temporels

Thomas CHATAIN

ENS Cachan, UniverSud, LSV
Av. Prés. Wilson, 94235 Cachan, France
Thomas.Chatain@ens-cachan.fr

Claude JARD

ENS Cachan, Université Européenne de Bretagne, IRISA
Campus de Ker-Lann, 35170 Bruz, France
Claude.Jard@bretagne.ens-cachan.fr

***Abstract*—On considère des réseaux de Petri colorés, à contraintes linéaires et pouvant posséder des arcs de lecture. Sur cette classe, on définit une sémantique concurrente en termes de processus d'ordre partiel permettant de garder explicite l'indépendance entre des tirs de transitions. L'ensemble des processus peut être représenté en utilisant la notion de dépliage symbolique. Nous montrons alors comment les réseaux de Petri temporels peuvent être codés dans ce modèle à l'aide d'une transformation syntaxique préservant la concurrence. Cette transformation permet de définir la notion de dépliage de réseaux de Petri temporels et d'en donner une représentation par préfixe fini [1] .**

Mots clés : Réseaux de Petri temporels, Dépliages, Réseaux de Petri Colorés, Préfixes

I. Introduction et état de l'art

Les réseaux de Petri sont connus de longue date comme un modèle de base pour représenter les comportements de systèmes "concurrents" avec une attention particulière sur les problèmes dus au parallélisme asynchrone dans les systèmes répartis. Ce type de modèle peut permettre d'aborder de façon précise, voire automatique, des questions de preuve de correction, de génération de tests, de diagnostic... La sémantique généralement utilisée pour calculer les comportements du modèle est une sémantique séquentielle dite "entrelacée" [1]. Plus récemment [2], l'effort a porté sur la définition de sémantique dites "concurrentes" (présentées souvent comme des dépliages). Celles-ci définissent les comportements du modèle comme étant des ordres partiels. Non seulement cela permet de réduire la combinatoire de la vérification dans le cas de parallélisme important, mais cela présente surtout à notre avis le mérite d'expliciter les dépendances causales dans les exécutions. Cette approche a conduit à de nouvelles applications pour le test et le diagnostic dans les grands systèmes répartis [3] [4]. Des questions plus difficiles sont apparues par la suite pour traiter non seulement des aspects fonctionnels, mais aussi des aspects non fonctionnels impliquant le temps notamment (les questions de contrôle de la qualité de service dans les réseaux par exemple). Théoriquement parlant, le mariage du temps et de la concurrence est un sujet difficile puisque la prise en compte de contraintes temporelles globales a tendance à casser la possibilité de parallélisme. Dans le contexte des réseaux de Petri temporels, la notion de dépliage n'a que été très récemment étudiée [5], [6] et a conduit à des sémantiques spécifiques.

Un autre point de vue, celui développé dans cet article, est de considérer le temps comme une façon de munir les jetons du réseau d'un age. Cet age peut être considéré comme une valeur portée par les jetons et donc a priori représenté par l'extension classique des réseaux de Petri connue sous le nom de réseaux colorés [7]. Dans un tel modèle, l'évolution des ages des jetons est définie par des contraintes symboliques posées sur les transitions. Nous avons donc considéré une classe particulière de réseaux colorés adaptée à la description des contraintes linéaires de progression du temps induites par la sémantique traditionnelle des réseaux temporels. La nouveauté présentée dans l'article consiste à munir de tels réseaux d'une sémantique concurrente, puis de montrer qu'en effet les réseaux temporels peuvent être codés dans ces réseaux colorés. Ce travail permet de mieux comprendre le lien entre données et temps et de retrouver une notion de dépliage pour les réseaux temporels. La classe des réseaux colorés choisie est assez large pour pouvoir construire des dépliages de différentes extensions temporelles des réseaux (avec arcs inhibiteurs, avec des paramètres, ...), ouvrant ainsi le champ d'application des dépliages à des modèles de plus grande complexité et plus réalistes vis-à-vis des phénomènes à capturer.

La suite de l'article présente les réseaux colorés considérés, puis leur sémantique concurrente et sa représentation par un dépliage symbolique. On rappelle ensuite la définition des réseaux temporels, pour montrer comment ils peuvent être codés en réseau coloré et conduire pour cette sous-classe particulière à l'existence de préfixes finis complets.

II. Réseaux de Petri colorés

Nous définissons une classe de réseaux de Petri colorés dans laquelle les places peuvent posséder un jeton au plus (réseau sauf), celui-ci portant une valeur réelle dans $\mathbb{R}$ (le domaine des couleurs est dense). Une transition peut consommer les jetons des places en entrée et produire des jetons dans les places de sortie. Elle peut aussi se contenter de lire les valeurs de jetons en entrée (par des arcs spéciaux dit de lecture). La sélection des valeurs des jetons est déterminée par une expression booléenne associée à la transition et portant sur

[1]Ce travail fait partie du projet national ANR DOTS sous la référence ANR-06-SETI-003.

des variables désignant les valeurs des jetons d'entrée et des variables (primées) désignant les valeurs des jetons de sortie.

A. Syntaxe

Commençons par donner quelques notations. Pour un ensemble A, on note $\mathcal{P}(A)$, l'ensemble de ses parties. Etant donné un ensemble de variables V à valeurs réelles, on considère l'ensemble des expressions Expr(V) formées par combinaison booléenne $(\wedge, \vee, \neg)$ de termes de la forme $x - y \bowtie c$ ou $x \bowtie c$ avec $x, y \in V$, $c \in \mathbb{R}$ et $\bowtie \in \{<, \leq, =, \geq, >\}$. L'ensemble V' désigne une copie de l'ensemble V dans laquelle tous les éléments v ont été primés, c'est-à-dire remplacés par v'. Lors du franchissement d'une transition, des variables v seront consommées ou lues, tandis que les variables v' seront écrites. Pour une expression ξ, on note $(\xi)[x/f(x)]_{x \in V}$ l'expression obtenue en remplaçant toutes les variables $x \in V$ par $f(x)$, f étant une fonction de renommage des variables.

Définition [syntaxe] : Un *réseau de Petri coloré* est donné par le n-uplet $(P, T, pre, post, cont, G, (M_0, \xi_0))$ où P est un ensemble fini de *places*, T est un ensemble fini de *transitions*, Pour une transition $t \in T$, $pre \in T \to \mathcal{P}(P)$ désigne ses places d'entrée et on notera cet ensemble ${}^\bullet t \stackrel{\text{def}}{=} pre(t) \subseteq P$. $post \in T \to \mathcal{P}(P)$ désigne ses places de sortie et on notera cet ensemble $t^\bullet \stackrel{\text{def}}{=} post(t) \subseteq P$. $cont \in T \to \mathcal{P}(P)$ désigne ses places de contexte (reliées en entrée de la transition par des arcs de lecture) et on notera cet ensemble $\bar{t} \stackrel{\text{def}}{=} cont(t) \subseteq P$. On impose de plus que le jeton d'une place d'entrée n'est jamais consommé et lu à la fois : ${}^\bullet t \cap \bar{t} = \emptyset$. On considérera qu'une transition a toujours au moins une place en entrée. $G \in T \to$ Expr$(P \cup P')$ désigne les gardes (ou contraintes symboliques) associées aux transitions. Elles ne peuvent faire apparaître que des variables locales à la transition, c'est-à-dire que $G(t) \in$ Expr$({}^\bullet t \cup \bar{t} \cup t^{\bullet\prime})$. Ces variables sont les noms des places connectées à la transition. Elles sont primées pour les places en sortie. $M_0 \in \mathcal{P}(P)$ est le marquage initial du réseau et $\xi_0 \in$ Expr(P) la contrainte initiale donnant les valeurs initiales possibles des places. Noter que dans les expressions, les noms des places marquées sont utilisés comme des variables pour désigner la couleur des jetons.

La figure 1 donne un exemple de réseau de Petri coloré, formé des places x et y, des transitions u, v, w. Les gardes sont écrites à côté des transitions et les ensembles $pre, post, cont$ sont implicitement définis par les arcs du graphe : la consommation et la production de jetons sont indiquées par des arcs orientés, la lecture est indiquée par un arc non orienté entre la place lue et la transition de lecture. Dans l'exemple, on prend $M_0 = \{x, y\}$ et $\xi_0 \equiv (x = 0) \wedge (y \geq 0)$. Noter que :

- la transition u incrémente strictement la valeur présente dans la place x,
- la transition v ne se déclenche que si la valeur présente dans la place y est négative et recopie alors dans la place y la valeur présente dans la place x,
- la transition w demande que la valeur dans la place y soit strictement positive et décrémente alors de 1 la valeur présente dans la place y.

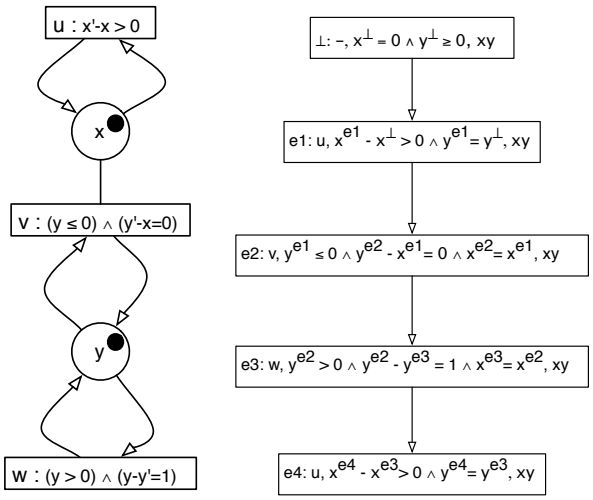

Fig. 1. Exemple de réseau de Petri coloré et une trace d'exécution symbolique (à droite)

B. Sémantique symbolique séquentielle

Notre modèle de réseau de Petri coloré peut être naturellement muni d'une sémantique opérationnelle dynamique en considérant le tir successif de transitions. Nous présentons ici les exécutions dans un formalisme très proche de la définition des processus qui sera vue plus loin. Chaque tir de transition t définit un nouvel événement e. Pour une place x, on notera x^e la variable mémorisant la valeur du jeton dans la place x après le tir de la transition désignée par l'événement e. A chaque événement e, on associe aussi la contrainte C_e liant les nouvelles valeurs des variables $(x^e)_{x \in M_e}$ aux valeurs précédentes. $M_e \subseteq P$ est le marquage produit par la transition.

Formellement, chaque événement e est défini par un quadruplet $({}^\bullet e, \tau_e, C_e, M_e)$ où ${}^\bullet e$ désigne l'événement précédent, $\tau_e \in T$ désigne la transition considérée.

L'ensemble des *traces* séquentielles est défini par un ensemble E d'événements.

On suppose que E contient un événement initial, noté $\bot$, correspondant à une transition fictive $-$. E forme un graphe dont les arcs sont définis par la relation de précédence immédiate, notée $e \to f \stackrel{\text{def}}{=} ({}^\bullet f = e)$. La fermeture transitive de cette relation $\to$ est notée $\to^*$. Par définition, les événements n'ont qu'un seul prédécesseur immédiat dans le graphe. Comme le graphe est sans cycle, il s'agit donc d'un arbre. Les différents chemins depuis la racine $\bot$ sont les comportements séquentiels du réseau, à condition que les contraintes données par les gardes soient satisfiables tout au long du chemin. Pour un événement $e \in E$, on note $\downarrow e \stackrel{\text{def}}{=} \{f \in E | f \to^* e\}$ l'ensemble des prédécesseurs de e. Ceci est formalisé par la définition qui suit.

Définition [traces séquentielles] : L'ensemble des traces symboliques séquentielles $\mathcal{T}$ est défini inductivement par (E est une trace, e est un événement) :

$\{\bot\} \in \mathcal{T}$, avec $\bot \stackrel{\text{def}}{=} (\emptyset, -, \xi_0[x/x^\bot]_{x \in M_0}, M_0)$;

$$\left\{\begin{array}{l} {}^\bullet e = max(E) \\ {}^\bullet\tau_e \cup \overline{\tau_e} \subseteq M_{\bullet_e} \\ C_e \stackrel{\text{def}}{=} \left\{\begin{array}{l} G(\tau_e)[x/x^{{}^\bullet e}]_{x\in {}^\bullet\tau_e \cup \overline{\tau_e}}[x'/x^e]_{x\in\tau_e{}^\bullet} \\ \bigwedge_{x\in M_{\bullet_e}\setminus {}^\bullet\tau_e}(x^e = x^{{}^\bullet e}) \end{array}\right. \\ \bigwedge_{f\in\downarrow e} C_f \text{ est satisfiable} \\ M_e = \tau_e{}^\bullet \setminus {}^\bullet\tau_e \end{array}\right\} \Rightarrow E\cup\{e\} \in \mathcal{T}$$

Noter que la condition d'activation d'un événement exprime que la garde de la transition associée doit être satisfiable en renommant simplement les variables consommées et lues en les indiçant par le dernier événement de la trace, ainsi que les variables écrites (celles primées) en les indiçant par l'événement que l'on cherche à placer (qui deviendra le dernier de la trace en cas de succès). Pour les variables qui ne sont pas touchées par la transition considérée, on indique que leur valeur reste identique en affirmant l'égalité de la variable précédente avec la variable courante. Pour qu'une transition soit tirable, il faut la condition usuelle sur les marquages, mais aussi vérifier que la nouvelle contrainte symbolique soit compatible avec toutes celles des événements de la trace. Si oui, la trace peut être prolongée par l'événement considéré. La fonction *max* considérée est au sens de la relation de causalité. Par construction $max(E)$ est un singleton (le dernier événement de la trace).

Les valeurs possibles des places après avoir parcouru une trace se terminant par un événement e sont définies par élimination des variables intermédiaires, c'est-à-dire celles définies par l'expression suivante :

$$(\exists x^f)_{f\in\downarrow{}^\bullet e, x\in M_f}\left(\bigwedge_{f\in\downarrow e} C_f\right)[x^e/x]_{x\in M_e}$$

La figure 1 à droite montre une trace du réseau de Petri de gauche, obtenue en considérant successivement le tir des transitions uvw. Les événements $({}^\bullet e, \tau_e, C_e, M_e)$ sont représentés par des rectangles dans lesquels figurent le nom de l'événement e, suivi du nom de la transition correspondante du réseau (τ_e), de la contrainte associée C_e et du marquage M_e obtenu. On dessine un arc de l'événement ${}^\bullet e$ à l'événement e. Il est à noter que le nom des événements n'est donné que par souci de lisibilité puisque dans la définition mathématique un événement contient l'ensemble de ses prédécesseurs.

L'union de l'ensemble des traces forme ce que l'on peut appeler un *dépliage symbolique séquentiel* du réseau. Cette union peut être représentée par un arbre permettant de partager les préfixes communs des traces. Sa définition inductive découle de la définition des traces en enlevant la condition de maximalité pour le rattachement d'un nouvel événement et en limitant la condition de satisfaction aux prédécesseurs causaux.

Définition [dépliage séquentiel] : Un dépliage séquentiel est un ensemble d'événements U défini inductivement par :
$\bot \in U$, avec $\bot \stackrel{\text{def}}{=} (\emptyset, -, \xi_0[x/x^\bot]_{x\in M_0}, M_0)$;

$$\left\{\begin{array}{l} {}^\bullet\tau_e \cup \overline{\tau_e} \subseteq M_{\bullet_e} \\ C_e \stackrel{\text{def}}{=} \left\{\begin{array}{l} G(\tau_e)[x/x^{{}^\bullet e}]_{x\in {}^\bullet\tau_e \cup \overline{\tau_e}}[x'/x^e]_{x\in\tau_e{}^\bullet} \\ \bigwedge_{x\in M_{\bullet_e}\setminus {}^\circ\tau_e{}^\bullet}(x^e = x^{{}^\bullet e}) \end{array}\right. \\ \bigwedge_{f\in\downarrow{}^\bullet e\cup\{e\}} C_f \text{ est satisfiable} \\ M_e = \tau_e{}^\bullet \setminus {}^\bullet\tau_e \end{array}\right\} \Rightarrow e \in U$$

Ce dépliage est en général infini (dès qu'il y a un cycle de comportement dans le réseau). Il n'y a pas d'espoir d'en produire une représentation finie par un préfixe complet vu la puissance théorique du modèle affirmée par le théorème suivant. Nous reconsidérerons cette question plus tard lorsque l'on codera les réseaux de Petri temporels avec nos réseaux de Petri colorés.

Théorème [puissance de Turing] : Le modèle de réseau de Petri coloré sauf considéré à la puissance d'une machine de Turing.

Preuve : Il suffit de coder le fonctionnement d'une machine à deux compteurs. Pour une place x, l'incrémentation consiste à franchir une transition contrainte par $x' - x = 1$, la décrémentation est mise en œuvre par la contrainte $x' - x = -1$ et le test à zéro directement par la contrainte $x = 0$ (le saut étant mis en œuvre par le contrôle fini du réseau sous-jacent).

III. Sémantique "concurrente"

La sémantique séquentielle précédente distingue par exemple les séquences $uvwu$ et $uvuw$ ne différant que par l'ordre de tir de deux transitions indépendantes. On peut vérifier d'ailleurs que la conjonction des conditions d'activation figurant dans chacune des traces définit le même ensemble de valeurs, à savoir celles satisfaisant l'expression $(x > 0) \wedge (y > -1)$. La sémantique concurrente a pour objectif de ne pas les distinguer et de représenter les exécutions par des ordres partiels au lieu de séquences totalement ordonnées. C'est ce que l'on appelle "processus" dans le vocabulaire des réseaux de Petri [8]. Cette notion est bien connue dans le cadre des réseaux de Petri ordinaires. Pour les processus symboliques des réseaux colorés, le point délicat est de définir les contraintes symboliques et les variables sur lesquelles elles portent.

Nous reprenons le même type de formalisme que pour les traces, en proposant une structure de graphe d'événements [9]. Chaque événement pourra cette fois-ci avoir plusieurs prédécesseurs. La notion de marquage est donc éclatée ("localisée"), le marquage donné dans chaque événement désigne les places qui ont été écrites par la transition associée. L'autre différence concerne les contraintes associées aux événements où il n'est plus demandé que les valeurs des variables non connectées à la transition ne bougent pas puisque la concurrence est maintenant représentée explicitement évitant ainsi l'introduction de la contrainte d'entrelacement des traces.

Un processus est défini par un ensemble d'*événements* E. Chaque événement $e \in E$ est un quadruplet $({}^\bullet e, \tau_e, C_e, M_e)$ qui code l'occurrence de la transition τ_e dans le processus. ${}^\bullet e$ est un ensemble d'événements précédant immédiatement l'événement e (les événements des transitions ayant écrits des places consommées ou lues par la transition considérée) : $f \rightarrow e \stackrel{\text{def}}{=} f \in {}^\bullet e$. La relation $\rightarrow^*$ est maintenant un ordre partiel (dite "relation de causalité"). M_e est l'ensemble des places écrites par la transition τ_e. On notera $M_E \stackrel{\text{def}}{=} \bigcup_{e\in E} M_e$, l'ensemble des places écrites par les événements de E et $M_E^{-1}(x) \stackrel{\text{def}}{=} \{y \in E \mid x \in M_y\}$ l'ensemble des événements de E écrivant la place x. On notera aussi par $\uparrow_E^x \stackrel{\text{def}}{=} max(M_E^{-1}(x))$

le dernier événement (au sens de la causalité) qui a écrit la place x. Cette notation est justifiée par le fait que $M_E^{-1}(x)$ est un ensemble totalement ordonné.

Comme précédemment, pour former les contraintes associées aux événements, on considère un jeu de variables $\{x^e\}$ donné par les places d'entrée x de la transition associée et les événements e précédant l'événement considéré.

Définition [processus] : L'ensemble des processus $\mathcal{P}$ est défini inductivement (E est un processus, e un événement) :
$\{\perp\} \in \mathcal{P}$, avec $\perp \stackrel{\text{def}}{=} (\emptyset, -, \xi_0[x/x^\perp]_{x\in M_0}, M_0)$;
$$\left.\begin{cases} {}^\bullet e = \bigcup_{x\in {}^\bullet\tau_e \cup \overline{\tau_e}} \{\uparrow_E^x\} \\ C_e \equiv G(\tau_e)[x/x^{\uparrow_E^x}]_{x\in {}^\bullet\tau_e\cup\overline{\tau_e}}[x'/x^e]_{x\in\tau_e{}^\bullet} \\ M_e = \tau_e{}^\bullet \\ \bigwedge_{f\in E\cup\{e\}} C_f \text{ est satisfiable} \end{cases}\right\} \Rightarrow E\cup\{e\}\in\mathcal{P}$$

Deux exemples de processus sont donnés dans la figure 2. Ils sont obtenus en considérant les suites de transitions $uvwu$ pour le premier et uv avec les transitions u et v en concurrence pour le deuxième.

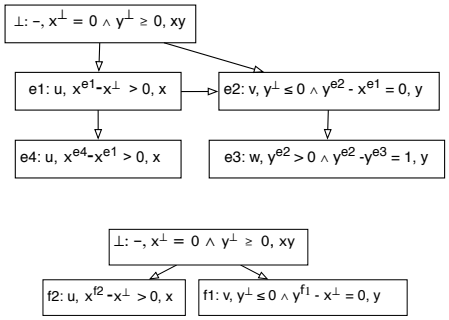

Fig. 2. Exemples de processus pour notre exemple.

Cette sémantique "concurrente" est équivalente à la sémantique séquentielle dans le sens où toute trace définit un processus et où tout ordre total des événements d'un processus définit une trace.

Théorème [: Correspondance traces et processus]
- Toute trace définit un processus ;
- Tout processus définit un ensemble de traces.

Preuve : Pour prouver la première assertion, considérons un trace séquentielle E, et la succession des transitions considérées $\sigma(E) \in T^*$ définie inductivement par :
- $\sigma(\emptyset) = \epsilon$ (le mot vide) ;
- $\sigma(E) = \sigma(E \setminus max(E)).\tau_{max(E)}$.

Il suffit alors de construire inductivement le processus E' en appliquant la définition III en considérant successivement les transitions de $\sigma(E)$. Encore faut-il montrer qu'elles sont tirables au sens de la sémantique concurrente. Considérons la trace formée de la succession des événements $e_0, ..., e_n$ (avec $e_0 = \perp$). La contrainte satisfiable associée à la trace est $\bigwedge_{i\in[0,n]} C_{e_i}$. L'idée est d'effacer récursivement les égalités $(x^{e_{i-1}} = x^{e_{i-2}})_{x\in M_{{}^\bullet e_{i-1}}\setminus {}^\bullet\tau_{e_{i-1}}}$ dans $C_{e_{i-1}}$ et de les propager dans C_{e_i} en le remplaçant donc par $C_{e_i}[x^{e_{i-1}}/x^{e_{i-2}}]_{x\in M_{{}^\bullet e_{i-1}}\setminus {}^\bullet\tau_{e_{i-1}}}$. On fait apparaître ainsi les contraintes associées aux événements des processus utilisant seulement les variables dernièrement écrites.

Pour prouver la seconde assertion, on considère un processus E. Il est défini par un ensemble d'événements partiellement ordonné par la relation de causalité $\rightarrow^*$. Considérons alors l'ensemble des extensions linéaires de cet ordre partiel. Soit $(E, <)$ une de ces extensions. Il faut montrer que cela forme une trace. On applique donc la définition II-B. Encore faut-il montrer que la succession des transitions est effectivement tirable au sens de la sémantique séquentielle. Il s'agit de la manipulation symbolique inverse de la précédente. On remplace dans C_{e_i} les variables x^{e_j} avec $j < i-1$ qui apparaissent par $x^{e_{i-1}}$ et on ajoute l'égalité $(x^{e_{i-1}} = x^{e_{i-2}})_{x\in M_{{}^\bullet e_{i-1}}\setminus {}^\bullet\tau_{e_{i-1}}}$ dans $C_{e_{i-1}}$ pour retrouver les contraintes de la trace.

IV. Dépliages symboliques

Les processus sont des ensembles d'événements. Le dépliage est tout simplement défini comme étant l'union de tous les processus du réseau de Petri.

Soit $\mathcal{P}$ l'ensemble des processus du réseau de Petri considéré. On note $U \stackrel{\text{def}}{=} \bigcup_{E\in\mathcal{P}} E$ son dépliage.

Graphiquement, ce dépliage est la superposition des graphes des processus. Deux nœuds sont superposés si ils ont les mêmes contenus au renommage près des événements (du fait du codage) et les mêmes prédécesseurs. La figure 3 montre la superposition des deux processus de la figure 2, formant alors un sous-ensemble de U que l'on appelle un *préfixe*.

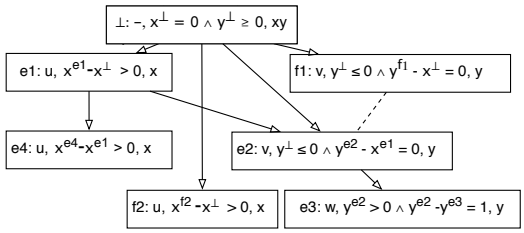

Fig. 3. Exemple de préfixe du dépliage de notre exemple. Le conflit entre événements est figuré en reliant les événements par des pointillés (dans ce cas simple, il est binaire).

Avant de donner une méthode inductive pour construire le dépliage, il est nécessaire de considérer la notion de *conflit* entre événements. Le conflit permet de noter qu'un ensemble d'événements du dépliage ne peuvent arriver ensemble dans un même processus. La donnée de l'ensemble ordonné U et de la relation de conflit est la structure d'événement asymétrique définissant la sémantique concurrente de nos réseaux de Petri.

Pour cela, on introduit la notion de *causalité faible* entre les événements e et f, notée $e \nearrow f$, pour dire que si un processus contient e et f, alors e arrive juste avant f. C'est-à-dire que $e \nearrow f$ ssi $(e \rightarrow f) \vee (\overline{\tau_e} \cap {}^\bullet\tau_f \neq \emptyset)$.

On dira qu'un ensemble d'événements F est en conflit, noté $\#F$, si il y en a qui consomment un même jeton (dans ce cas par construction, ils ne peuvent appartenir au même processus) ou si la causalité faible crée un cycle dans cet ensemble.

$$\#F \equiv \left\{\begin{array}{l} \exists e, f \in F \,:\, e \neq f \wedge {}^\bullet\tau_e \cap {}^\bullet\tau_f \neq \emptyset \\ \exists e_0, e_1, ..., e_n \in F \,:\, e_0 \nearrow e_1 \nearrow ... \nearrow e_n \nearrow e_0 \end{array}\right. \vee$$

Définition [dépliage] : Le dépliage U d'un réseau est défini inductivement par :

$$\perp \in U, \text{ avec } \perp \stackrel{\text{def}}{=} (\emptyset, -, \xi_0[x/x^{\perp}]_{x\in M_0}, M_0) \text{ ;}$$
$$\left.\begin{array}{l} {}^{\bullet}\tau_e \cup \overline{\tau_e} = M_{{}^{\bullet}e} \wedge \neg\#(\downarrow {}^{\bullet}e \cup \{e\}) \\ C_e \equiv G(\tau_e)[x/x^{\uparrow^{x}_{\downarrow^{\bullet}e}}]_{x\in {}^{\bullet}\tau_e \cup \overline{\tau_e}}[x'/x^{e}]_{x\in \tau_e{}^{\bullet}} \\ M_e = \tau_e{}^{\bullet} \\ \bigwedge_{f\in\downarrow e} C_f \text{ est satisfiable} \end{array}\right\} \Rightarrow e \in U$$

Noter que la définition du dépliage ressemble à la définition des processus. La différence est que les nouveaux événements ne se raccrochent pas forcément sur des événements terminaux. En contrepartie, il est demandé que le passé causal de l'événement rattaché soit sans conflit. On peut vérifier sur notre exemple que la considération successive des transitions $uvwuv$ produit le préfixe du dépliage montré dans la figure 3.

V. Réseaux de Petri temporels (RdPT)

Définition [syntaxe] : Un *réseau de Petri temporel* est donné par le n-uplet $(P, T, pre, post, efd, lfd, M_0)$ où P est un ensemble fini de *places* et T est un ensemble fini de *transitions*. Pour une transition $t \in T$, $pre \in T \to \mathcal{P}(P)$ désigne ses places d'entrée et on notera cet ensemble ${}^{\bullet}t \stackrel{\text{def}}{=} pre(t) \subseteq P$. $post(t) \in T \to \mathcal{P}(P)$ désigne ses places de sortie et on notera cet ensemble $t^{\bullet} \stackrel{\text{def}}{=} post(t) \subseteq P$. $M_0 \in \mathcal{P}(P)$ est le marquage initial du réseau. $efd : T \to \mathbb{R}$ est la fonction qui, à chaque transition, définit la date de tir au plus tôt de la transition. La fonction $lfd : T \to \mathbb{R} \cup \{\infty\}$ définit la date de tir au plus tard de la transition.

Dans la représentation graphique des réseaux temporels, l'intervalle fermé $[efd(t), lfd(t)]$ ou l'intervalle semi-ouvert $[efd(t), \infty)$ quand $lfd(t) = \infty$ est écrit à côté de chaque transition (voir la figure 4).

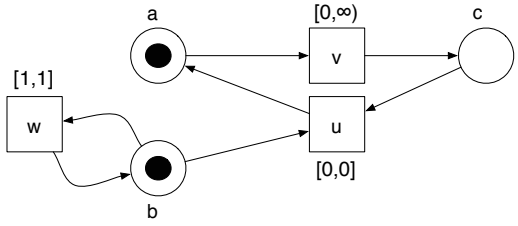

Fig. 4. Exemple de réseau de Petri temporel.

La sémantique standard est séquentielle. L'état du réseau temporel est donné par une paire (M, dob) où $M \subseteq P$ est le marquage classique, et $dob : M \to \mathbb{R}$ définit les dates de naissance des jetons dans chaque place.

Soit 0 la fonction nulle. L'état initial du réseau est défini par $(M_0, 0)$. Une transition $t \in T$ est tirable à la date θ dans l'état (M, dob) et conduit à l'état (M', dob') si et seulement si ses places d'entrée sont marquées (${}^{\bullet}t \subseteq M$) et

- l'attente minimum est effectuée à partir du moment où la transition est sensibilisée : $max_{p\in {}^{\bullet}t} dob(p) + efd(t) \leq \theta$
- aucune transition activable n'aura dépassé son attente maximum à la date de tir de la transition : $\forall t' \in T, {}^{\bullet}t' \subseteq M \Rightarrow \theta \leq max_{p\in {}^{\bullet}t'} dob(p) + lfd(t')$
- le temps progresse : $\theta \geq max_{p\in P} dob(p)$

Pour pouvoir dater la dernière action par $max_{p\in P} dob(p)$, on suppose que $\forall t \in T, t^{\bullet} \neq \emptyset$. L'état atteint après le tir de t est $((M \setminus {}^{\bullet}t) \cup t^{\bullet}, dob')$ où $dob'(p) = dob(p)$ si $p \in M \setminus {}^{\bullet}t$ et $dob'(p) = \theta$ si $p \in t^{\bullet}$.

L'exemple de la figure 4 engendre le langage de transitions $w^*v[w]uv$ dans lequel on peut commencer par faire un nombre quelconque de w suivi de v et d'au plus un w, avant de terminer par u puis v. Noter bien la différence par rapport au langage du réseau non temporisé dans lequel w peut être répété un nombre quelconque de fois après la première occurrence de v.

VI. Codage des RdPT en réseaux colorés

A. Sémantique entrelacée

La première idée est d'utiliser les valeurs réelles des places pour représenter les dates de naissance des jetons, puisque dans les réseaux temporels, le fait qu'une transition t soit tirable dépend de la date d'arrivée des jetons dans ses places d'entrée (puisqu'il faut attendre $efd(t)$ après que la transition ait été sensibilisée). Mais sa tirabilité dépend aussi des dates d'arrivée des jetons dans d'autres places du réseau (celles qui sensibilisent les transitions dans le marquage global courant de la sémantique séquentielle), puisqu'il faut assurer qu'aucune transition t' sensibilisée du réseau dépasse son $lfd(t')$. Ces dépendances supplémentaires seront représentées par des arcs de lecture puisqu'elles ne se traduisent pas par la consommations de jetons.

Le problème est que marquage global de la sémantique séquentielle est une notion dynamique. Pour définir une transformation syntaxique, nous devons considérer l'ensemble des possibilités d'ajout d'arcs de lecture. Ce qui conduit à dupliquer les transitions. Soit $PE(t)$ l'ensemble des ensembles de places d'entrée possibles pour la transition t du réseau temporel. Ce peut être a priori n'importe quel ensemble de places incluant les places d'entrée de la transition et pouvant être marqué. Pour éviter des duplications inutiles pour lesquelles on est sûr que les places d'entrée ne peuvent pas être marquées ensemble, il est utile de considérer les flots de places du réseau de Petri sous-jacent au réseau temporel. Pour cela, les fonctions pre et $post$ peuvent être vues comme des matrices booléennes sur $P \times T$ et les marquages comme des vecteurs booléens sur P. Remarquons d'abord qu'un marquage M du réseau de Petri temporel ne sera accessible à partir du marquage initial M_0 que si il est déjà accessible dans le réseau de Petri simple sous-jacent. Considérons une suite $\sigma \in T^*$ de transitions franchissables du réseau. Le marquage obtenu obéit à l'équation $M = M_0 + (post - pre)\overline{\sigma}$, où $\overline{\sigma}$ est un vecteur sur $\mathbb{N}^{|T|}$ comptant le nombre de transitions de chaque type dans la séquence σ. On peut alors considérer les vecteurs sur le places $\phi \in \{0,1\}^{|P|}$ tels que $\phi.(post - pre) = 0$. D'après l'équation sur les marquages, les solutions de base forment des invariants $\phi.M = \phi.M_0$ indiquant l'exclusivité entre places du réseau. Soit

$$PE(t) = \{L \subseteq P \mid {}^{\bullet}t \subseteq L \wedge \forall \phi \in \{0,1\}^{|P|}$$

tel que $\phi.(post - pre) = 0, \phi.L \leq \phi.M_0\}$ les différents marquages partiels pouvant potentiellement sensibiliser la transition t.

Conformément à la sémantique des réseaux temporels, la contrainte symbolique du marquage initial est $\xi_0 \equiv \bigwedge_{x \in M_0}(x = 0)$.

Dans un premier temps on va considérer une transformation préservant le caractère séquentiel de la sémantique. On considère alors seulement les marquages partiels maximaux. L'ensemble des transitions du réseau coloré est $\bigcup_{t \in T}\{(L,t) \mid L$ est maximal dans $PE(t)\}$. Pour toute transition $\tau \stackrel{\text{def}}{=} (L,t)$, on note ${}^\bullet\tau \stackrel{\text{def}}{=} {}^\bullet t$, $\overline{\tau} \stackrel{\text{def}}{=} L \setminus {}^\bullet t$ et $\tau^\bullet \stackrel{\text{def}}{=} (L \setminus {}^\bullet t) \cup t^\bullet$.

Conformément à la sémantique séquentielle des réseaux temporels, la garde $G(\tau)$ associée à la transition τ est définie par :

$$G((L,t)) \equiv \bigwedge_{q \in t^\bullet} \begin{cases} (max_{p \in {}^\bullet t} p + efd(t) \leq q' \wedge \\ \bigwedge_{p \in t^\bullet, p \neq q}(p' = q') \wedge \\ \bigwedge_{u \mid {}^\bullet u \subseteq L}(q' \leq max_{p \in {}^\bullet u} p + lfd(u)) \wedge \\ max_{p \in L} p \leq q' \end{cases}$$

En pratique, on ne gardera que les transitions τ pour lesquelles $G(\tau)$ est satisfiable. Les conditions de la garde sont la recopie pure et simple des conditions de la sémantique standard séquentielle. Cette transformation, qui ne fait apparaître aucune concurrence, a le mérite de coder les entrelacements des transitions en explicitant les dépendances par des arcs de lecture. Il faut bien noter que les gardes proposées sont des expressions valides pour les réseaux colorés : la fonction maximum utilisée n'est qu'un raccourci d'écriture qui peut être ramené à une expression booléenne. L'utilisation de variables de places non connectées en entrée des transitions du réseau temporel implique l'ajout d'arcs de lecture dans le réseau coloré correspondant.

Dans notre exemple de la figure 4, les places a et c sont exclusives et on a : $PE(u) = \{bc\}, PE(v) = \{a, ab\}, PE(w) = \{b, ab, bc\}$. On considère donc les 4 transitions suivantes : $\{(bc,u),(ab,v),(ab,w),(bc,w)\}$. On obtient après simplification les contraintes suivantes :

$$\begin{array}{lcl} G((bc,u)) & = & (max(b,c) = a') \wedge (a' \leq b+1) \\ G((ab,v)) & = & (a \leq c') \wedge (c' \leq b+1) \wedge (max(a,b) \leq c') \\ G((ab,w)) & = & (b' = b+1) \wedge (max(a,b) \leq b') \\ G((bc,w)) & = & (b' = b+1) \wedge (max(b,c) = b') \end{array}$$

Le résultat de la transformation est donné dans la figure 5.

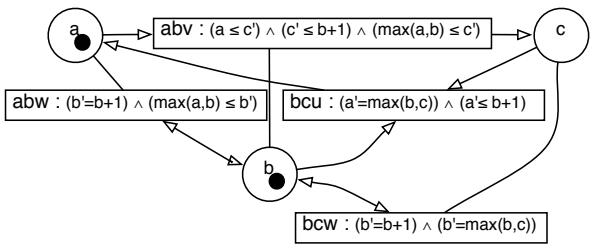

Fig. 5. Exemple de réseau coloré séquentiel équivalent au réseau temporel de la figure 4.

Ce réseau est équivalent du point de vue séquentiel au réseau temporel. On voit que, partant de la situation $a = b = c = 0$, on peut répéter la transition abw engendrant dans la place b les entiers naturels. Après avoir tiré x transitions, on aura $a = 0, b = x, c = 0$. La transition bcw n'était par contre pas tirable à cause de la lecture de la place c demandée. La transition abv peut être ensuite tirée, conduisant à l'état $a = 0, b = x, c \in [x, x+1]$. Dans cette situation, la transition bcw peut être à nouveau tirée et conduit à $a = 0, b = x+1, c \in [x, x+1]$, empêchant alors de la tirer une deuxième fois. Ce réseau par contre ne possède plus de concurrence puisque les transitions abv et abw sont en conflit (à cause de la cyclicité de la causalité faible), de même pour les transitions abv et bcw. Dans le réseau temporel initial, les transitions v et w étaient pourtant en concurrence à l'instant 1 notamment. C'est l'objet du paragraphe suivant de montrer comment on peut réinjecter de la concurrence malgré l'existence d'un temps global.

B. Sémantique concurrente

La sémantique concurrente va autoriser à considérer le tir de transitions même si l'ordre temporel n'est pas respecté. On va, d'une part, relâcher la contrainte de maximalité de L et d'autre part, relâcher la contrainte de progression linéaire du temps.

Sans la contrainte de maximalité de L, il faut augmenter l'ensemble des transitions à considérer, ensemble des transitions pouvant influer sur les dates possibles de tir de la transition. Ce n'est plus l'ensemble des transitions u sensibilisées dans le marquage global, mais l'ensemble des transitions susceptibles de consommer un jeton dans les places de L (c'est-à-dire les transitions u telles que ${}^\bullet u \cap L \neq \emptyset$) et qui peuvent être potentiellement sensibilisées en même temps que t (c'est-à-dire telles que $\exists L' \in PE(t) \mid {}^\bullet u \cup L = L'$). Les autres sont indépendantes et ne peuvent pas interférer avec le tir de la transition t considérée. On remplace donc dans la garde la condition ${}^\bullet u \subseteq L$ par $({}^\bullet u \cap L \neq \emptyset) \wedge (\exists L' \in PE(t) \mid {}^\bullet u \cup L = L')$. Pour ce qui concerne la contrainte du progression du temps, cela ne doit pas se faire sans précaution. Dans notre exemple de la figure 1, si on enlevait simplement la contrainte de progression, on obtiendrait notamment la garde $G((ab,w)) = (b' = b+1)$ impliquant que la transition w peut être répétée même après que v est tirée, exécution pourtant illégale dans le réseau temporel. Le problème, pour permettre à des parties différentes du réseau d'évoluer de façon non synchronisée dans le temps, est de veiller à la cohérence causale des places partagées en lecture.

L'information des dates de naissances des jetons n'est plus suffisante puisque les jetons des places lues peuvent ne pas être consommés. On propose de mettre dans les places p une information supplémentaire qui est la date $\overline{p}$ de la dernière lecture du jeton.

Pour celà, chaque place p du réseau temporel est représentée par un couple de places $(p, \overline{p})$ dans le réseau coloré. La duplication des places s'effectue par la transformation suivante permettant la mise à jour des places $\overline{p}$ lorsque la place p est lue par une transition :

- Lorsque la place p est en sortie d'une transition t, la place $\overline{p}$ l'est aussi.
- Lorsque la place p est lue par la transition t, la place $\overline{p}$ doit être consommée puis écrite par la transition t.

Ces connexions supplémentaires apparaissent dans les contraintes rajoutées dans les gardes.

Pour résumer, on propose la construction des transitions suivantes, pour chaque $t \in T$ et chaque $L \in PE(t)$:

$$G((L,t)) \equiv$$
$$\bigwedge_{q \in t^\bullet} \begin{cases} (max_{p \in {}^\bullet t} p + efd(t) \leq q') \wedge \bigwedge_{p \in t^\bullet, p \neq q} (p' = q') \wedge \\ \bigwedge_{u | ({}^\bullet u \cap L \neq \emptyset) \wedge (\exists L' \in PE(t) | {}^\bullet u \cup L = L')} \\ \qquad (q' \leq max_{p \in {}^\bullet u \cap L} p + lfd(u)) \wedge \\ max_{p \in L} \overline{p} \leq q' \wedge (\overline{q}' = q') \wedge \bigwedge_{p \in L \setminus {}^\bullet t} (\overline{p}' = q') \end{cases}$$

En pratique, on éliminera les redondances éventuelles en ne gardant que les transitions (L,t) telles que $G((L,t)) \wedge \bigwedge_{L' \subset L} \neg G((L',t))$ est satisfiable.

Dans notre exemple de la figure 4, on obtient les gardes suivantes :

$$\begin{array}{lcl} G((bc,u)) & = & (max(\underline{b},\underline{c}) = a') \wedge (a' \leq b+1) \wedge \\ & & (max(\overline{b},\overline{c}) \leq a') \wedge (\overline{a}' = a') \\ G((a,v)) & = & (a = c') \wedge (\overline{a} \leq c') \wedge (\overline{c}' = c') \\ G((ab,v)) & = & (a \leq c') \wedge (c' \leq b+1) \wedge (max(\overline{a},\overline{b}) \leq c') \wedge \\ & & (\overline{c}' = c') \wedge (\overline{b}' = c') \\ G((b,w)) & = & faux \\ G((ab,w)) & = & (b' = b+1) \wedge (max(\overline{a},\overline{b}) \leq b') \wedge \\ & & (\overline{b}' = b') \wedge (\overline{a}' = b') \\ G((bc,w)) & = & (b' = b+1) \wedge (b' \leq max(b,c)) \wedge \\ & & (max(\overline{b},\overline{c}) \leq b') \wedge (\overline{b}' = b') \wedge (\overline{c}' = b') \end{array}$$

Les transitions correspondant aux gardes $G((a,v))$ et $(G((b,w))$ peuvent être ignorées. Le résultat de la transformation est donné dans la figure 6.

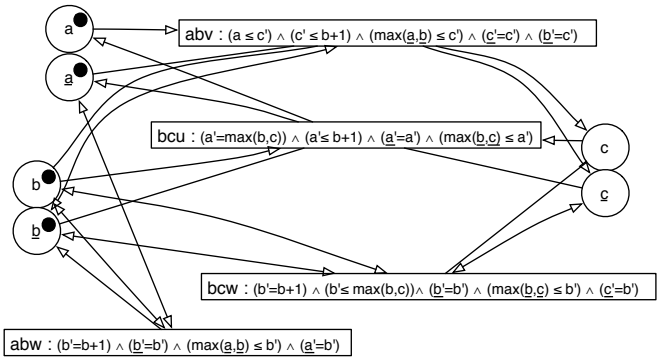

Fig. 6. Exemple de réseau coloré concurrent équivalent au réseau temporel de la figure 4. Les places supplémentaires $\overline{p}$ sont notées en souligné dans la figure.

On vérifie qu'initialement la transition abw est tirable et peut se répéter produisant la suite des entiers dans b et $\overline{b}$. Les transitions abv et abw peuvent être maintenant concurrentes car le cycle de causalité faible du codage de la figure 5 a été cassé par l'introduction des places $\overline{p}$ (la place a n'est plus en lecture de la transition abw).

La figure 7 montre un préfixe du réseau de la figure 6, lui-même codage du réseau temporel de la figure 4.

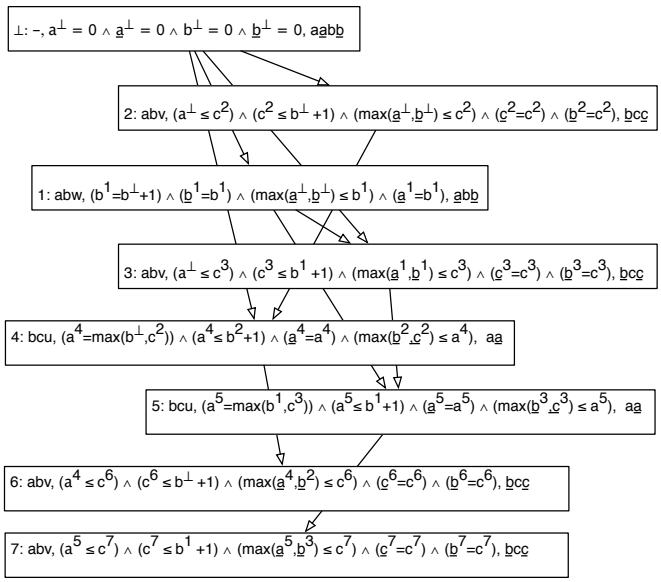

Fig. 7. Un préfixe du dépliage du réseau coloré codant le réseau temporel de la figure 4.

VII. Préfixes finis

La question est d'obtenir une représentation finie du dépliage. Plus exactement nous cherchons une condition d'arrêt dans l'algorithme du dépliage telle que le préfixe fini obtenu à terminaison contient suffisamment d'information pour décider de l'accessibilité d'un état donné par un marquage et une valuation des variables. Il était illusoire de trouver une telle condition pour les réseaux colorés généraux. La situation est différente pour les réseaux colorés codant les réseaux temporels. Il faut donc étudier la forme particulière des contraintes symboliques apparaissant dans le dépliage.

L'idée est de considérer les sous-ensembles d'événements clos par précédence causale et sans conflit. Ils représentent des processus. On considère alors la conjonction de toutes les contraintes des événements (cette contrainte globale est satisfiable par construction du dépliage). Ces contraintes peuvent être réécrites pour ne tomber que sur un nombre fini de contraintes possibles. On pourra alors arrêter le dépliage lorsqu'aucun nouvel événement ne permet plus d'obtenir de processus donnant une nouvelle contrainte globale. On retrouve ainsi la preuve de décidabilité du "model-checking" des RdPT saufs [10].

Théorème [Préfixe fini] : L'ensemble des contraintes suivantes associées aux processus E est fini. De plus, deux processus qui donnent la même contrainte peuvent être prolongés par les mêmes exécutions.

$$(\exists x^f)_{f \in E, x \in M_f} \begin{cases} \bigwedge_{e \in E} C_e \wedge \\ \bigwedge_{x \in M} x = \max\{\theta - x^{\uparrow_E^x}, C\} \wedge \\ \bigwedge_{u \in T, {}^\bullet u \subseteq M} \theta - \max_{x \in {}^\bullet u} x^{\uparrow_E^x} \leq lfd(u) \end{cases}$$

où M désigne le marquage atteint à la fin du processus, $\theta \stackrel{\text{def}}{=} \max_{x \in M} x^{\uparrow_E^x}$ est la date atteinte et C est la plus grande constante (finie) intervenant dans les intervalles temporels du

réseau. La dernière ligne vérifie qu'aucune transition sensibilisée dans l'état final n'a dépassé sa date de tir au plus tard.

Preuve : On se souvient d'abord que le nombre de marquages est fini car le réseau est sauf. Les variables de type x^f étant éliminées par la quantification existentielle, les contraintes ne portent que sur les variables $x \in M$, qui représentent ici l'âge des jetons dans l'état final. Les contraintes sont des combinaisons booléennes d'inégalités entre termes de type $x + c$ avec $x \in P$ et $c \in \mathbb{N}$. De plus le maximum avec la plus grande constante du réseau C permet de ne conserver que des constantes inférieures à C. On obtient donc un nombre fini d'expressions.

Comme les exécutions possibles à partir d'un état ne dépendent que du marquage et de l'âge des jetons, et comme le vieillissement des jetons n'a plus d'influence à partir de C, deux processus qui donnent la même contrainte peuvent être prolongés par les mêmes exécutions.

VIII. Conclusion et perspectives

Cet article a défini la notion de dépliage symbolique pour des réseaux de Petri colorés à valeurs réelles et à contraintes linéaires. Différentes applications habituelles des dépliages deviennent alors accessibles pour cette nouvelle classe très expressive de modèles des systèmes répartis. Nous avons par ailleurs montré que ce type de réseau symbolique peut coder les contraintes temporelles des réseaux de Petri. Comme perspective immédiate, on peut reprendre la démarche pour étudier des extensions existantes des réseaux temporels. Par exemple, on peut considérer les réseaux à chronomètres [11] dans lesquels des arcs inhibiteurs peuvent être attachés à des transitions temporelles. Le chronomètre associé à la transition pour compter la durée de sensibilisation sera arrêté si un jeton est présent dans une des places reliées à la transition par un arc inhibiteur.

On propose de coder un chronomètre (qui est en train de compter) avec deux variables (c'est-à-dire des places) :

- la date θ_l de dernier déclenchement,
- la durée d qui avait déjà été comptée par le chronomètre avant θ_l.

La valeur affichée à la date $\theta \geq \theta_l$ s'exprime comme $d + (\theta - \theta_l)$. Cette proposition utilise des expressions un peu plus générales que les $x - y = c$. Ici il faut pouvoir additionner (ou soustraire) deux variables. Cela étend les contraintes linéaires pour définir de façon plus générale des polyèdres, mais c'est un cadre dans lequel la satisfaction des contraintes reste décidable.

Du coup, on peut aisément penser à traiter des modèles paramétrés dans lesquels les bornes temporelles des transitions peuvent être considérées comme des variables libres. On reste dans le cadre de contraintes décidables. Une application possible dans le domaine de la supervision des systèmes répartis par exemple serait, à partir d'une suite d'observations d'événements, inférer non seulement les dépendances causales qui expliquent les observations, mais aussi trouver les contraintes reliant les paramètres du modèle (suivant l'idée exprimée dans [12] dans le cadre du modèle des réseaux d'automates temporisés). Nous pensons que le cadre symbolique choisi est bien adapté au paramètrage des modèles, réponse intéressante au problème de la conception de modèles robustes. Nous avons souvent constaté la difficulté de positionner les constantes du modèle. Donner la possibilité d'en garder sous une forme de paramètre et laisser à l(outil d'analyse le soin de trouver les plages de fonctionnement réaliste nous semble un réel progrès dans l'utilisation d'une démarche fondée sur des modèles formels.

References

[1] M. Diaz, *Les réseaux de Petri - Modèles Fondamentaux*. Hermes, 2001.

[2] J. Esparza and K. Heljanko, *Unfoldings, A Partial-Order Approach to Model Checking*, ser. Monographs in Theoretical Computer Science. Springer, 2008.

[3] C. Jard, "Synthesis of distributed testers from true-concurrency models of reactive systems," *Information & Software Technology*, vol. 45, no. 12, pp. 805–814, 2003.

[4] A. Benveniste, E. Fabre, C. Jard, and S. Haar, "Diagnosis of asynchronous discrete event systems, a net unfolding approach," *IEEE TAC*, vol. 48.

[5] T. Aura and J. Lilius, "A causal semantics for time petri nets," *Theor. Comput. Sci.*, vol. 243, no. 1-2, pp. 409–447, 2000.

[6] T. Chatain and C. Jard, "Complete finite prefixes of symbolic unfoldings of safe time petri nets," in *ICATPN*, 2006, pp. 125–145.

[7] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured petri nets and cpn tools for modelling and validation of concurrent systems," *STTT*, vol. 9, no. 3-4, pp. 213–254, 2007.

[8] J. Engelfriet, "Branching processes of Petri nets," *Acta Informatica*, vol. 28, no. 6, pp. 575–591, 1991.

[9] P. Baldan, A. Corradini, and U. Montanari, "An event structure semantics for p/t contextual nets: Asymmetric event structures," in *Proceedings of FoSSaCS '98*. Springer Verlag, 1998, pp. 63–80.

[10] D. Lime and O. H. Roux, "Model checking of time petri nets using the state class timed automaton," *Discrete Event Dynamic Systems*, vol. 16, no. 2, pp. 179–205, 2006.

[11] B. Berthomieu, D. Lime, O. H. Roux, and F. Vernadat, "Reachability problems and abstract state spaces for time petri nets with stopwatches," *Discrete Event Dynamic Systems*, vol. 17, no. 2, pp. 13–158, 2007.

[12] B. Grabiec and C. Jard, "Unfolding of networks of automata and their application in supervision," in *Proceedings of NOTERE*, 2009.