Sur la synthèse de la commande des systèmes à événements discrets temporisés

Alexandru Tiberiu Sava

Institut National Polytechnique de Grenoble

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : **≪AUTOMATIQUE-PRODUCTIQUE≫**

préparée au Laboratoire d'Automatique de Grenoble

dans le cadre de l'École Doctorale **≪Électronique, Électrotechnique, Automatique, Télécommunication et Signal≫**

présentée et soutenue publiquement

par

Alexandru Tiberiu SAVA

le 23 novembre 2001

Titre :

SUR LA SYNTHÈSE DE LA COMMANDE DES SYSTÈMES À ÉVÉNEMENTS DISCRETS TEMPORISÉS

Directeur de thèse :

M. Hassane ALLA (Laboratoire d'Automatique de Grenoble)

JURY :

Président	M. Jean Michel DION	Directeur de recherche CNRS
Rapporteur	M. Jean Louis FERRIER	Professeur LISA-ISTIA, Université d'Angers
Rapporteur	M. Guy JUANOLE	Professeur Université Paul Sabatier
Examinateur	Mme. Simona CARAMIHAI	Professeur Université Politehnica de Bucarest
Examinateur	M. Olivier ROUX	Professeur Ecole Centrale de Nantes
Examinateur	M Hassane ALLA	Professeur Université Joseph Fourier

A Oana,
A mes parents,
A moi même aussi,
(à mon avis, je mérite ...)

Remerciements

Le travail de recherche présenté dans ce mémoire a été réalisé au Laboratoire d'Automatique de Grenoble, sous la coordination de M. Hassane ALLA, professeur à l'Université Joseph Fourier.

Je remercie chaleureusement à mon professeur M. Hassane ALLA, qui a été beaucoup plus qu'un directeur de thèse.

Je tiens à remercier M.Jean Michel DION, directeur de recherche CNRS, qui m'a fait l'honneur de présider le jury réuni pour la soutenance de ma thèse de doctorat.

Je remercie également M. Jean Louis FERRIER, professeur à l'Université d'Angers et à M.Guy JUANOLE, professeur à l'Université Paul Sabatier, pour leurs commentaires précis et judicieux, et pour avoir accepté la charge d'être rapporteurs de mes travaux.

J'exprime toute ma gratitude à Mme Simona CARAMIHAI, professeur à l'Université "Politehnica" de Bucarest, qui m'a dirigé vers la recherche scientifique.

Je remercie à M. Olivier ROUX, professeur à l'Ecole Centrale de Nantes, qui lui aussi m'a fait l'honneur d'avoir accepté de porter un jugement sur mon travail de recherche et de faire partie du jury de soutenance de ma thèse.

Je garderai un bon souvenir de ces années que j'ai passées au Laboratoire d'Automatique de Grenoble. Je remercie le directeur M. Luc DUGARD de m'avoir accueilli dans ce laboratoire.

Je tiens tout particulièrement à remercier mes amis Antonio FAVELA CONTERAS, Monika KUROVSKI, Rosa ABBOU, Catalin CIOCOIU, Anca et Frederic MAYER, Costin ENE, Moez YEDDES et Jean Marc BOLLON pour leur soutien moral et les bons moments que nous avons passé ensemble.

Mes remerciements s'adressent également aux membres des équipes $SYLODI$ et CS^2 dont j'ai fait partie durant cette période. Je remercie M. René DAVID, Mme. Maria DI MASCOLO, Mme. Mireille JACOMINO, Mme. Zineb SIMEU ABAZI, M. Christian COMMAULT et M. Pierre LADET pour leur sympathie, leurs remarques et conseils.

Je n'oublierai pas l'ensemble de membres de l'équipe administrative et technique du Laboratoire d'Automatique de Grenoble, dont je remercie pour leur gentillesse, leur bon humeur et leur efficacité.

Un grand merci à tous ce qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Merci enfin à Oana et à mes parents, pour ... tout.

Table des matières

Table des figures

Table des notations

p_i	place d'un RdP T-temporel
T_j	transition d'un RdP T-temporel
$\mathcal{M}$	ensemble des marquages d'un RdP T-temporel
m_1	marquage d'un RdP T-temporel
$\mathcal{L}$	ensemble des sommets d'un automate temporisé
L_i	sommet d'un automate temporisé
$\mathcal{T}$	ensemble des transitions d'un l'automate
$T_{m,n}$	transition de l'automate temporisé du sommet L_m vers le sommet L_n
$g_{m,n}$	garde associée à la transition $T_{m,n}$
$A_{m,n}$	affectation associée à la transition $T_{m,n}$
n	compteur qui mémorise le nombre des sommets d'un automate temporisé
$e_{m,n}$	espace des horloges à l'entrée dans le sommet L_n par le franchissement de la transition $T_{m,n}$, pour la visite courante
$E_{m,n}$	espace des horloges à l'entrée dans le sommet L_n par le franchissement de la transition $T_{m,n}$, pour toutes les visites
$E^d_{m,n}$	espace des horloges désiré à l'entrée dans le sommet L_n par le franchissement de $T_{m,n}$
E_n	espace des horloges à l'entrée dans le sommet L_n
E^a_n	l'espace des horloges actives à l'entrée dans le sommet L_n
E^d_n	espace des horloges désiré à l'entrée dans le sommet L_n
D_n	espace des horloges actives désiré dans le sommet L_n
$g^d_{m,n}$	la nouvelle garde calculée pour la transition $T_{m,n}$
$S_{m,n}$	espace des horloges actives dans le sommet L_m qui vérifient la garde $g_{m,n}$ de la transition de sortie $T_{m,n}$
$S^d_{m,n}$	espace des horloges actives dans le sommet L_m qui vérifient la nouvelle garde $g^d_{m,n}$ de la transition de sortie $T_{m,n}$
Q	ensemble des sommets à partir desquels il faut actualiser l'automate
F	ensemble des sommets interdits
P	une pile qui mémorise les visites des sommets non encore analysées

Introduction

Un système à événements discrets (SED) est un système dynamique dont l'espace d'état est discret. L'évolution de ces systèmes est déterminée par l'occurrence instantanée des événements. Ainsi, tant qu'il n'y a pas d'occurrence d'un événement, l'état du système reste inchangé. Ces systèmes se retrouvent dans des nombreux domaines d'application tels que les systèmes de production, l'informatique, les réseaux de communication, etc.

Depuis quelques années, les exigences d'une compétitivité sans cesse croissante ainsi que le progrès technologique ont engendré l'apparition des systèmes de production de plus en plus complexes. Compte tenu des enjeux économiques, on ne peut pas se permettre la synthèse d'une commande par des essais et des corrections successives. Ainsi, là où dans le passé le bon sens suffisait, il est devenu crucial de disposer des outils formels et des techniques pour l'analyse et la synthèse de la commande qui permettent de garantir, a priori, que le fonctionnement du système commandé respecte les spécifications imposées par le cahier des charges.

Les bases de la théorie de la supervision des SED ont été établies par les travaux de Ramadge et Wonham, dans les années 80, [RW87] [RW89]. Dans ces travaux, on prend en compte seulement la succession logique des événements. Le temps n'intervient pas explicitement. Ainsi, l'étude se place à un niveau qualitatif. Les outils employés sont les modèles automates et langages formels.

Un procédé est considéré comme étant un générateur spontané d'événements. Son fonctionnement est caractérisé par un ensemble de séquences des événements.

Un superviseur est un SED qui permet de modifier le fonctionnement d'un procédé afin d'éviter toute évolution vers un état non désiré, i.e. qui ne vérifie pas les spécifications imposées. Son rôle est de restreindre l'ensemble de séquences d'événements qui peuvent être générées par le procédé en interdisant l'occurrence de certains événements.

Dans la théorie de Ramadge et Wonham, le rôle du superviseur est limité à autoriser ou interdire l'occurrence des événements. Il n'est pas habilité à forcer le procédé à générer des événements. Par conséquent, cette théorie ne peut pas être utilisée pour la commande des SED.

Pour répondre à ce problème, des travaux de recherche ont été effectués pour étendre la théorie de la supervision des SED à celle de la commande supervisée. Nous citons les travaux de Charbonnier [Cha96] qui utilise le concept d'événement forçable pour développer une approche de synthèse de la commande supervisée des SED. Ce concept désigne une catégorie d'événements qui peuvent se produire spontanément ou être forcés par un système extérieur. Dans ce cas, un superviseur a le rôle d'un système de commande qui peut forcer l'occurrence de certains événements selon l'état courant du procédé.

Les approches proposées pour la commande supervisée des SED permettent d'élaborer des lois de commande efficaces, qui garantissent le respect des spécifications imposées par le cahier des charges. Cependant, ces approches ne prennent pas en compte explicitement l'influence du temps. Elles considèrent que les événements peuvent avoir lieu dans

un procédé à des moments indéterminés, donc n'importe quand. Dans la réalité, le fonctionnement de la plupart des processus industriels est sujet à des contraintes temporelles telles que les dates de lancement des tâches, et les durées opératoires. Par conséquent, on dispose d'une information supplémentaire sur la date d'arrivée des événements qui peut être utilisée pour élaborer des lois de commande moins contraignantes. A titre d'exemple, on étudiera le cas particulier d'un système où la synthèse pour le modèle autonome donne un superviseur vide, alors qu'il y a une solution pour le modèle temporisé. Cette idée a été développée dans plusieurs approches de synthèse de la commande des SEDT.

L'approche proposée dans les travaux de Brandin et Wonham [BW94] étend la théorie de la supervision des SED par l'introduction des événements forçables ainsi que des contraintes temporelles associées aux dates d'occurrence d'événements. La prise en compte du passage du temps est matérialisée par l'occurrence d'un événement particulier. Par conséquent, on obtient un modèle à temps discret. Cette approche fournit des bonnes solutions pour la commande des systèmes à événements discrets temporisés (SEDT). Cependant, dans la plupart des cas, la nature discrète du temps engendre l'explosion combinatoire du nombre d'états.

Pour contourner le problème de l'explosion du nombre d'états, plusieurs auteurs ont développé des approches d'analyse et synthèse de la commande pour des SEDT basées sur l'outil automate temporisé [Gou99] [MPS95] [AMPS98]. Ces approches ont les inconvénients de modélisation spécifiques aux modèles automates. Ces outils ne permettent pas de représenter explicitement certains mécanismes fréquemment rencontrés dans le fonctionnement d'un SEDT, tels que la synchronisation, le parallélisme ou le partage des ressources.

Un autre outil de modélisation et analyse très utilisé pour l'étude des SED est le modèle réseau de Petri [DA92]. Par opposition à l'automate, cet outil fournit une représentation naturelle et intuitive des SED.

Une approche de synthèse de la commande des SEDT qui associe la capacité de modélisation des réseaux de Petri avec la puissance d'analyse des automates a été proposée dans [Kou99]. Le système à commander est modélisé par un réseau de Petri à arcs temporels. Ensuite, on modélise le comportement de ce réseau de Petri avec un automate à temps continu. La synthèse de la commande est effectuée à partir de cet automate. La méthode proposée par cette approche s'adresse au cas particulier où les variables qui modélisent l'influence du temps ne sont pas couplées.

D'autres travaux sur la synthèse d'un superviseur pour un SEDT sont présentés dans [Car97] et [CA97].

L'étude des SEDT est effectué à la fois par deux communautés différentes : les informaticiens et les automaticiens. Ces communautés exploitent ces systèmes pour répondre à des objectives différents. D'un coté, les informaticiens s'intéressent principalement à la vérification des spécifications. D'un autre coté, l'objectif des automaticiens est la synthèse de la commande des SEDT.

L'objectif du travail présenté dans ce mémoire est le développement d'une méthodologie de synthèse de la commande qui se propose de répondre au problème dans le cas général.

Dans un SEDT, les transitions entre états se font à l'occurrence des événements dans des intervalles de temps déterminés. L'analyse permettra de caractériser l'espace d'état atteignable pour une condition initiale donnée. Il peut s'avérer qu'une partie de cet espace ne soit pas désirée si elle contredit les spécifications. Dans ce cas, l'étape de synthèse consistera à remettre en cause certaines conditions temporelles pour sur l'occurrence des événements pour modifier l'espace atteignable et le faire coïncider avec l'espace désiré.

L'approche que nous proposons pour la synthèse de la commande des SEDT comporte

les étapes suivantes :

- On représente le système à commander par un réseau de Petri T-temporel ;
- On modélise le comportement du réseau de Petri T-temporel associé au procédé par un automate temporisé ;
- La synthèse de la commande est basée sur des résultats de la théorie des automates temporisés concernant l'analyse d'atteignabilité des états.

L'objectif du premier chapitre de ce mémoire est de positionner notre travail par rapport aux approches existant dans la littérature. D'abord nous faisons une brève incursion dans la théorie de la supervision des SED. Ensuite, nous concentrons notre attention sur les principales approches de synthèse de la commande des SEDT. Nous présentons leurs principes et leurs limites pour la synthèse de la commande pour des systèmes complexes.

Le deuxième chapitre est dédié à la présentation des outils de modélisation et analyse des SEDT que nous avons retenu dans nos travaux de recherche. Nous insistons sur les outils réseau de Petri T-temporel et automate temporisé que nous utilisons dans notre approche de synthèse de la commande d'un SEDT. Nous détaillons également la première étape de cette approche, qui consiste à représenter le SEDT à commander par un modèle réseau de Petri T-temporel.

Nous présentons, dans le troisième chapitre, l'algorithme que nous proposons pour construire l'automate temporisé qui modélise le comportement d'un réseau de Petri T-temporel. Cette méthode est une des contributions de cette thèse. Elle nous donne la possibilité d'utiliser les techniques puissantes d'analyse d'atteignabilité spécifiques aux automates temporisés pour analyser les propriétés des systèmes modélisés par des réseaux de Petri T-temporels. Nous utilisons cette méthode dans la deuxième étape de la synthèse de la commande, qui est une étape d'analyse. La construction de l'automate temporisé qui modélise le comportement du réseau de Petri associé permet de déterminer les évolutions non-désirées possibles dans le fonctionnement du procédé.

Le quatrième chapitre de ce mémoire introduit la méthode que nous avons développé pour la synthèse de la commande. Notre objectif est de trouver l'ensemble de toutes les lois de commande qui garantissent que le fonctionnement du système respecte les spécifications imposées par le cahier des charges. Ce chapitre constitue la partie la plus importante de notre travail.

Enfin, nous achevons ce mémoire par une conclusion sur ce qui a été fait et des perspectives de continuation de ce travail.

Chapitre 1

Sur la commande par supervision des SEDT

Dans ce chapitre, une introduction sur la commande par supervision des systèmes à événements discrets temporisés (SEDT) est faite. D'abord nous présentons le principe de la théorie de la supervision pour des systèmes à événements discrets (SED) initiée par Ramdge et Wonham. Puis nous faisons un rappel sur les approches de synthèse de la commande par supervision des SEDT proposées dans la littérature qui sont en relation directe avec nos travaux de recherche.

1.1 Introduction sur les SEDT

Un système à événements discrets (SED) est un système dynamique dans lequel l'espace des états est discret. Ses trajectoires d'états sont constantes par morceaux. Un tel système évolue conformément à l'occurrence des événements physiques à des intervalles de temps généralement irréguliers ou inconnus.

Considérons, par exemple, une machine qui peut être dans trois états : *arrêt*, *marche* et *panne*. On suppose qu'il peut y avoir l'occurrence de quatre événements : *début traitement*, *traitement achevé*, *panne* et *réparation*. Ces événements sont étiquetés par les symboles a, b, c et d respectivement. Une évolution possible de ce système est présentée dans la figure 1.1.

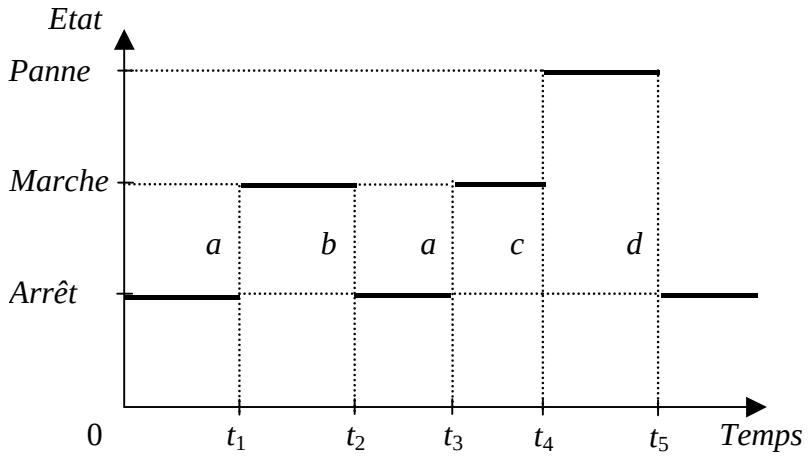


FIG. 1.1 – Evolution de l'état de la machine

Initialement la machine est en état d'*arrêt*. Suite à l'arrivé de l'événement a à l'instant t_1, la machine passe dans l'état *marche*. De façon similaire, le système évolue aux instants t_2, t_3, t_4 et t_5 sur l'occurrence des événements b, a, c et d.

Les domaines d'application des SED sont nombreux. Différents aspects du comportement d'un système peuvent être considérés, selon l'application envisagée. Par conséquent, différents outils pour la modélisation et l'analyse des SED ont été développés. L'évolution d'un SED est caractérisée par l'occurrence des événements. Selon la manière de modéliser l'arrivée des événements, on peut classifier les modèles des SED en trois catégories : modèles logiques, modèles temporisés non stochastiques et modèles temporisés stochastiques.

Les modèles logiques des SED (automates, réseaux de Petri autonomes) prennent en compte seulement l'ordre d'occurrence des événements. Ces modèles ignorent les instants d'occurrence des événements. Dans ce contexte, l'évolution de la machine présentée dans la figure 1.1 est décrit par la séquence : *abacd*... Ces modèles sont utilisés pour l'étude des propriétés qualitatives des SED.

L'analyse d'un SED représenté par un modèle logique commence par la spécification de l'ensemble des trajectoires d'états admissibles, i.e. les séquences d'événements physiques possibles dans le fonctionnement du SED. Considérons maintenant une propriété désirable pour le comportement du SED. L'objectif de l'analyse est de déterminer si chaque trajectoire d'état satisfait la propriété désirée. Par contre, la synthèse de la commande par supervision cherche à restreindre, par la commande, l'ensemble des trajectoires d'état admissibles au sous-ensemble de trajectoires d'état qui satisfont la propriété désirée. Des

exemples des propriétés désirables sont : utilisation correcte des ressources (exclusion mutuelle), enchaînement correct des événements, absence des blocages, etc.

Dans certaines applications, l'information temporelle est essentielle et doit être prise en compte explicitement par le modèle. Les modèles qui ont cette caractéristique sont appelés temporisés. Il s'agit des modèles où le temps est déterministe (réseaux de Petri temporels, automates temporisés) et des modèles où le temps est aléatoire (chaînes de Markov, réseaux des files d'attente).

Les modèles où le temps est déterministe ont le même principe de fonctionnement que les modèles logiques. Cependant, les instants d'occurrence des événements sont explicitement pris en compte. Les SED représentés par ces modèles sont appelés systèmes à événements discrets temporisés (SEDT).

Les modèles où le temps est aléatoire sont caractérisés par des distributions de probabilité associées aux instants d'arrivée des événements.

Dans la suite, nous nous intéressons seulement à des modèles logiques et modèles où le temps est déterministe.

1.2 Supervision des SED

La théorie de la supervision des SED a été initiée par les travaux de Ramadge et Wonham [RW87] [RW89]. Cette théorie utilise le modèle automate et les langages formels [HU79] pour modéliser le comportement d'un SED ainsi que les spécifications imposées pour son fonctionnement.

Le principe de la supervision d'un procédé est d'interdire l'occurrence de certains événements dans chacun de ses états et d'autoriser l'occurrence des autres par l'action d'un superviseur. L'objectif est de construire un superviseur tel que le procédé supervisé évolue avec un maximum de liberté tout en respectant les spécifications imposées pour son fonctionnement. Le schéma général du principe de la supervision est présenté dans la figure 1.2.

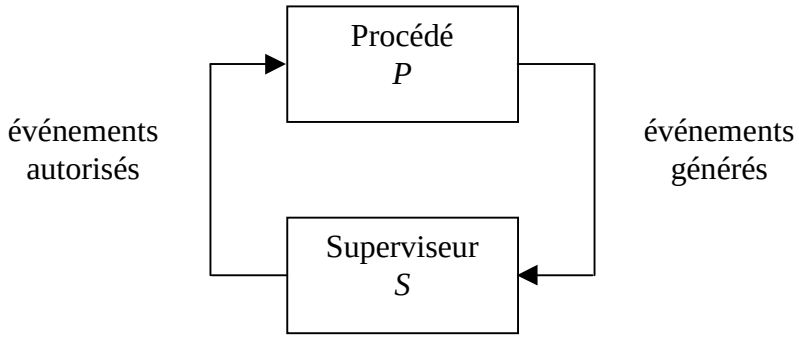


FIG. 1.2 – Schéma du principe général de la supervision

Le superviseur observe l'état du procédé par l'intermédiaire de la séquence des événements générés par le fonctionnement du procédé. Cette séquence est l'entrée du superviseur. En réponse, il agit sur le comportement du procédé par l'intermédiaire des lois de contrôle qui définissent les événements autorisés depuis l'état courant du procédé.

Généralement, un superviseur ne peut pas agir sur tous les événements qui interviennent dans le fonctionnement d'un procédé. En pratique, certains événements ne peuvent pas être interdits. Ces événements sont appelés incontrôlables. Les événements

qui peuvent être interdits quel que soit l'état du procédé sont appelés contrôlables. Par conséquent, l'ensemble $\sum$ est partagé en deux sous-ensembles disjoints : $\sum = \sum_c \cup \sum_u$. Le sous-ensemble $\sum_c$ mémorise les événements contrôlables tandis que $\sum_u$ dénote le sous-ensemble des événements incontrôlables. Dans le domaine de l'automatique, les événements contrôlables correspondent en général aux actions appliquées au procédé tandis que les événements incontrôlables sont associées aux sorties (fournies par les capteurs) du procédé.

Une entrée de contrôle pour un procédé P est un sous-ensemble $\gamma \subseteq \sum$ tel que $\sum_u \subseteq \gamma$. Elle représente l'ensemble des événements autorisés par le superviseur depuis un état du procédé. Etant donné que les événements incontrôlables ne peuvent pas être interdits, chaque entrée de contrôle contient tous les événements incontrôlables. L'ensemble des entrées de contrôle est noté avec Γ.

Formellement, un superviseur est défini par la fonction :

$$S : L(P) \to \Gamma$$

Le fonctionnement du procédé P non supervisé, modélisé par un langage $L(P)$, est appelé fonctionnement en boucle ouverte. Par contre, le fonctionnement du procédé couplé avec son superviseur S, modélisé par le langage $L(S/P)$, est appelé fonctionnement en boucle fermée. Ce fonctionnement doit respecter les spécifications imposées par le cahier des charges.

Définition 1.1. Le langage $L(S/P)$, qui décrit le fonctionnement d'un procédé P supervisé par un superviseur S est défini par :

- $e \in L(S/P)$,
- $wa \in L(S/P) \Leftrightarrow w \in L(S/P),\ a \in S(w)$ et $wa \in L(P)$

■

Un mot wa peut être généré par le procédé supervisé seulement si le mot w a été généré par le procédé supervisé, si l'événement a est autorisé par le superviseur et le mot wa est accepté par le procédé non supervisé. Le mot vide e est compris dans le langage $L(S/P)$. Ainsi, $L(S/P)$ est inclus dans le langage $L(P)$ et il est préfixe-clôt. Par conséquent, un superviseur ne peut que restreindre le comportement d'un procédé.

Soit $L_m(P)$ le langage marqué du procédé non supervisé. Le langage marqué du procédé couplé avec son superviseur, est $L_m(S/P) = L(S/P) \cap L_m(P)$. Ce langage décrit les tâches qui sont toujours accomplies en présence de la supervision.

On a vu que la tâche d'un superviseur est de générer l'entrée de contrôle tel que le procédé évolue avec un maximum de liberté tout en respectant certaines contraintes. Par contre, à cause de l'existence des événements incontrôlables, il n'est pas possible de restreindre le comportement du procédé à n'importe quel sous-ensemble.

On appelle langage désiré d'un procédé, l'ensemble de mots générés par le procédé tout en respectant les contraintes imposées par le cahier des charges. Soit K le langage désiré pour un procédé P. L'étude de l'existence d'un superviseur S tel que $L(S/P) = K$ est basée sur la propriété de contrôlabilité du langage K.

Définition 1.2. Un langage K est dit contrôlable par rapport à un langage $L(P)$ si $\overline{K} \sum_u \cap L(P) \subseteq \overline{K}$.

■

Si un langage K est contrôlable par rapport à un langage $L(P)$, alors sa clôture préfixielle, notée $\overline{K}$, est invariante par rapport à l'occurrence des événements incontrôlables.

Si un langage K n'est pas contrôlable par rapport à un langage $L(P)$, alors il n'existe aucun superviseur S tel que $L(S/P) = K$. Dans ce cas, il faut chercher un langage

contrôlable et préfixe-clôt inclus dans K. L'objectif étant de construire un superviseur le plus permissif possible, on cherche le plus grand langage préfixe-clôt et contrôlable, inclus en K, noté $SupC(K)$.

Le problème de l'incontrôlabilité d'un langage est illustré dans la figure 1.3.

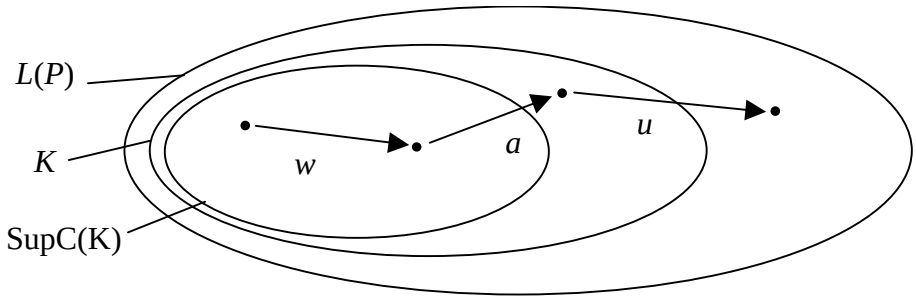


FIG. 1.3 – Incontrôlabilité d'un langage

Soit un mot $wau \in L(P)$, où :

- w est un mot tel que $w \in SupC(K)$,
- a est un événement contrôlable tel que $wa \in K - SupC(K)$,
- u est une séquence d'événements incontrôlables telle que $wau \in L(P) - K$.

Le langage K est incontrôlable parce qu'il est impossible d'interdire l'exécution du mot u à la suite de wa. Par contre, en interdisant l'occurrence de l'événement contrôlable a depuis le mot w, le problème de contrôlabilité ne se pose plus.

Les conditions d'existence d'un superviseur sont spécifiées dans la propriété suivante [RW87] :

Proposition 1.2.1. *Soit un SED P non bloquant de langage $L(P)$ et de langage marqué $L_m(P)$.*

1. *Pour tout langage $K \subseteq L(P)$ il existe un superviseur S tel que $L(S/P) = K$ ssi K est préfixe-clôt et contrôlable par rapport à $L(P)$.*
2. *Pour tout langage $K \subseteq L_m(P)$ il existe un superviseur S tel que $L_m(S/P) = K$ et le système en boucle fermée est non bloquant ssi K est L_m-fermé (c'est à dire $\overline{K} \cap L_m = K$) et contrôlable par rapport à $L(P)$.*

■

Cette proposition est essentielle dans la théorie de la supervision des SED, mais elle n'est pas utilisée pour la construction d'un superviseur. Dans la pratique, la synthèse du superviseur est effectuée à l'aide de l'outil automate. Le comportement du procédé à superviser ainsi que les spécifications imposées pour son fonctionnement sont modélisés par des automates. Le langage désiré du procédé est représenté par l'automate obtenu par la composition synchrone des automates qui modélisent le procédé et les spécifications.

Généralement, la propriété de clôture préfixielle du langage désiré d'un SED est vérifiée a priori. Par opposition, la contrôlabilité ne l'est pas nécessairement. Ainsi, la construction d'un superviseur commence par la recherche du plus grand sous-langage contrôlable inclus dans le langage désiré pour le procédé.

Plusieurs approches ont été proposées pour la supervision des SED.

Ramadge et Wonham [WR87] ont proposé un algorithme itératif pour la construction du plus grand sous-langage contrôlable inclus dans un langage donné.

Kumar et Garg [Kum91] ont proposé un algorithme pour la construction du superviseur le plus permissif basé sur la détermination des états interdits du modèle automate

représentant le fonctionnement désiré. Un état est considéré interdit s'il ne respecte pas les spécifications. C'est à dire, il est atteint depuis un état du procédé par le franchissement d'une transition de sortie sur un événement incontrôlable qui n'est pas autorisé par la spécification. Un état à partir duquel le système peut évoluer vers un état interdit par l'occurrence d'un événement incontrôlable est considéré faiblement interdit. Le superviseur est construit en éliminant de l'automate du fonctionnement désiré, tous les états interdits et faiblement interdits. Cet algorithme permet d'obtenir un superviseur le plus permissif.

Exemple 1.1. Considérons le poste de collage présenté dans la figure 1.4.

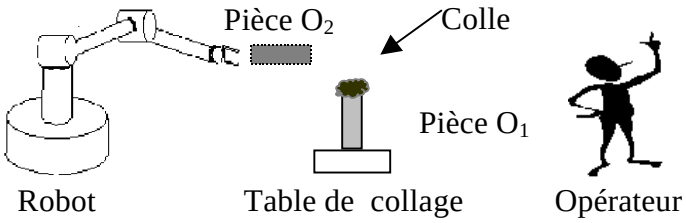


FIG. 1.4 – Poste de collage

Le problème à résoudre consiste à coller ensemble deux pièces, O_1 et O_2 tout en respectant certaines spécifications. Cet exemple sera traîté tout au long de ce mémoire.

La pièce O_1 est préparée par un opérateur qui dépose de la colle sur O_1. L'opérateur peut décider de l'instant de démarrage de sa tâche, donc cet événement, noté o, est contrôlable.

La pièce O_2 est transportée par un robot et déposée sur la pièce O_1 afin de réaliser le collage. Le démarrage de la tâche du robot, modélisé par l'événement r, est contrôlable. Par contre, l'accomplissement de cette tâche, représenté par l'occurrence de l'événement f, est incontrôlable.

La réalisation d'un collage de qualité est conditionnée par la consistance de la colle. Ainsi, elle ne doit être ni trop humide, ni trop sèche. Le fait que la colle soit devenue prête pour le collage est modélisé par l'occurrence de l'événement b. Par contre, l'occurrence de l'événement s modélise le fait que la colle est devenue trop sèche. Les événements b et s sont incontrôlables.

Par conséquent, l'ensemble des événements contrôlables est $\sum_c = \{o,\ r\}$ tandis que l'ensemble des événements incontrôlables est $\sum_u = \{f,\ b,\ s\}$.

On impose la spécification suivante sur le fonctionnement du procédé : il faut que le robot dépose la pièce O_2 pour le collage après que la colle soit devenue prête, mais avant qu'elle soit trop sèche. Ainsi, l'événement f doit avoir lieu après l'événement b, mais avant l'arrivée de l'événement s.

Le fonctionnement du poste de collage est modélisé par l'automate $\mathcal{A}_P$ représenté dans la figure 1.5.

L'état initial de cet automate est q_0. Un collage réussi est modélisé par l'état marqué q_7. Chacun des états $q_3,\ q_8,\ q_9,\ q_{10}$ modélise un collage raté. A partir de ces états, le système ne peut plus évoluer vers un état marqué.

La spécification imposée sur le fonctionnement du procédé peut être représentée par l'automate $\mathcal{A}_{Spec}$ illustré dans la figure 1.6. C'est la solution la plus simple, où on n'a pas besoin d'une représentation détaillée du fonctionnement du procédé.

Le modèle du comportement désiré du procédé est obtenu par la composition synchrone des automates $\mathcal{A}_P$ et $\mathcal{A}_{Spec}$. Cet automate est représenté dans la figure 1.7.

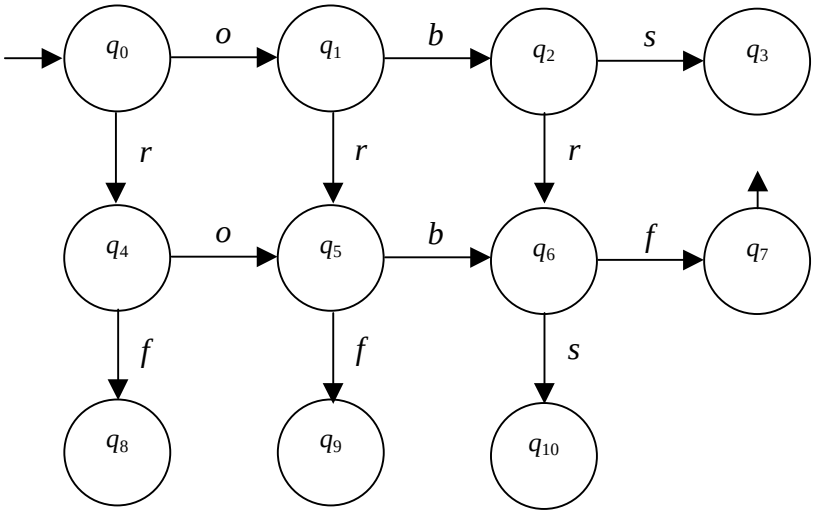


FIG. 1.5 – Le modèle du poste de collage

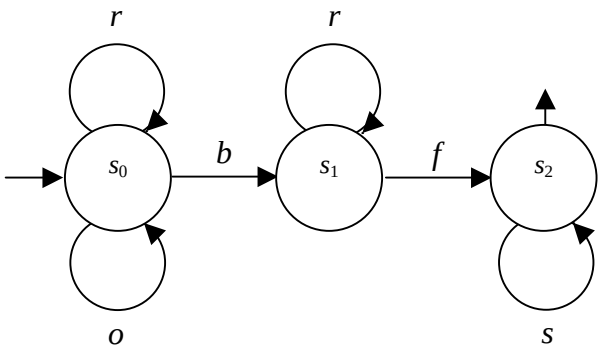


FIG. 1.6 – Modèle des spécifications

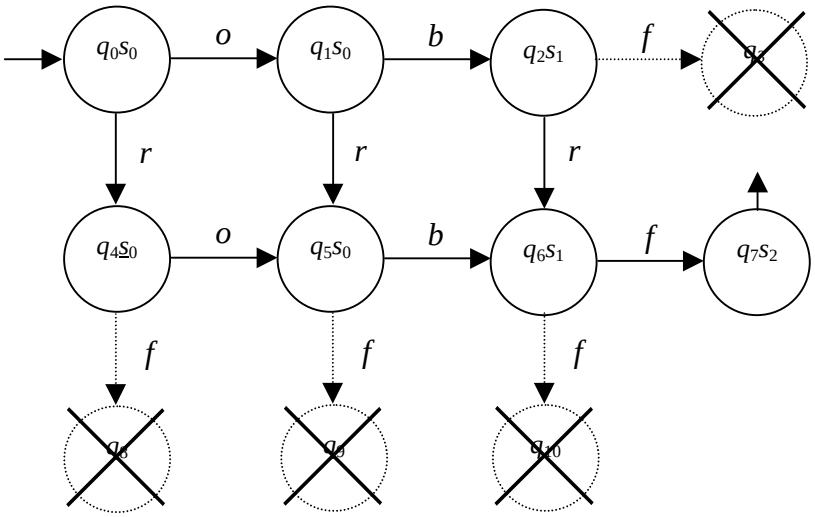


FIG. 1.7 – Modèle du comportement désiré

Les états q_3, q_8, q_9 et q_{10} sont des états interdits parce que l'évolution dans ces états est possible dans el procédé, alors qu'elle n'est pas autorisé par les spécifications.

L'état $q_2 s_1$ est faiblement interdit parce que l'occurrence de l'événement incontrôlable

s depuis cet état permet d'atteindre un état qui ne respecte pas la spécification imposée sur le comportement du procédé (un état interdit). Pour des raisons similaires, les états q_4s_0, q_5s_0, q_6s_1 sont eux aussi faiblement interdits.

L'utilisation de l'algorithme du Kumar pour la construction du superviseur consiste à éliminer de l'automate $\mathcal{A}$ tous les états qui sont interdits ou faiblement interdits.

L'automate du superviseur est présenté dans la figure 1.8. Cet automate a un seul état, q_0s_0.

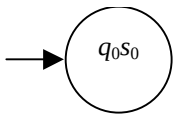


FIG. 1.8 – Automate du superviseur

Le même résultat est obtenu par la méthode proposée par Ramadge et Wonham [RW87]. D'abord on calcule le langage L_P de l'automate $\mathcal{A}_P$ qui modélise le comportement du procédé.

$$\begin{aligned} L(P) &= o(b(s + r(f + s + e) + e) + r(b(f + s + e) + f + e) + e) \\ &\quad + r(o(b(f + s + e) + f + e) + f + e) \end{aligned}$$

Le langage de l'automate qui modélise le comportement désiré est :

$$K = obrf + orbf + robf$$

Considérons le mot $ob \in \overline{K}$ et l'événement $s \in \sum_u$. Il faut noter que $obs \in \overline{K} \sum_u \cap L(P)$, mais $obs \notin \overline{K}$. Par conséquent, $\overline{K} \sum_u \cap L(P) \not\subseteq \overline{K}$, donc le langage K n'est pas contrôlable. Dans ce cas, il n'existe aucun superviseur S tel que $L(S/P) = K$. Le plus grand sous-langage contrôlable inclus dans le langage K est :

$$SupC(K) = e$$

Alors, il existe un superviseur S qui garantit que le procédé n'évolue jamais vers un état interdit.

$$L(S/P) = e$$

Ce langage correspond à l'automate illustré dans la figure 1.8.

Analysons maintenant l'existence d'un superviseur marqué pour le processus de collage, i.e. qui garantit un collage réussi. Le langage marqué de l'automate $\mathcal{A}_P$ est :

$$L_m(P) = robf + orbf + obrf$$

Cependant, $L_m(S/P) = L(S/P) \cap L_m(P) = \phi$, donc il n'y a aucun superviseur qui garantit un collage réussi.

Nous verrons par la suite qu'en prenant en compte les données temporelles concernant la date d'occurrence des événements, il y a une solution à ce problème.

■

La théorie de Ramadge et Wonham permet d'obtenir de bons résultats, mais son application à des systèmes complexes peut devenir difficile, voir impossible dû au grand nombre d'états. Certaines extensions de la théorie de base de Ramadge et Wonham ont été proposées pour réduire la complexité des problèmes de supervision. Il s'agit de la supervision modulaire, supervision sous observation partielle et supervision décentralisée [RW89]. Nous ne présentons pas ici ces approches.

1.3 Commande par supervision des SEDT

Jusqu'à présent nous avons présenté les notions principales concernant la supervision et la commande par supervision des SED à l'aide des modèles qui prennent en compte seulement l'ordre d'occurrence des événements. Dans ces modèles, on suppose implicitement que la date d'occurrence d'un événement se trouve dans l'intervalle $[0, \infty)$, c'est à dire que l'événement peut avoir lieu n'importe quand. Ceci n'est bien sûr pas vrai dans la réalité. Par exemple, la fin d'une opération aura lieu dans un intervalle dont la largeur exprimera l'incertitude sur la durée de l'opération : [3.1, 3.7]. Il faut alors pouvoir disposer de ces durées. Par conséquent, la modélisation de ces systèmes en vue de la synthèse de la commande par supervision doit prendre en compte explicitement l'influence du temps. Cette information peut être utilisée pour synthétiser des lois de commande moins contraignantes dans le sens où on peut restreindre la contrainte temporelle sur l'occurrence d'un événement sans l'interdire totalement. Cependant, la prise en compte du passage du temps rend les modèles étudiés plus complexes et la synthèse de la commande par supervision devient plus difficile.

Plusieurs travaux ont été élaborés et des approches basées sur différents outils de modélisation (automates, réseaux de Petri) ont été proposées pour l'analyse et la commande par supervision des SEDT. Par la suite, nous présentons une synthèse des travaux qui ont une relation directe avec l'étude présentée dans ce mémoire.

1.3.1 Commande par supervision en temps discret

Dans [BW94] les auteurs proposent une approche qui étend la théorie de Ramadge et Wonham par l'introduction du concept d'événement forcé, ainsi que des contraintes temporelles associées aux dates d'occurrence des événements.

Cette approche utilise deux types d'automates pour modéliser le comportement d'un SED. D'abord, on modélise les états du système et les transitions entre ces états par un automate $\mathcal{A}_{act} = (Q_{act}, \sum_{act}, \delta_{act}, q_{act,0}, Q_{act,m})$ où :

- Q_{act} est un ensemble fini d'états logiques ;
- $\sum_{act}$ est un ensemble fini d'événements ;
- δ_{act} est la fonction de transition ;
- $q_{act,0}$ est l'état initial ;
- $Q_{act,m}$ est l'ensemble des états marqués.

Chaque événement du procédé est instantané et il est exécuté à n'importe quel instant t du temps réel. Cependant, on suppose que le temps est mesuré à l'aide d'une horloge digitale qui incrémente un compteur de top d'horloge défini par :

$$top : \mathbb{R}^+ \to \mathbb{N}, \text{ tel que } top(t) = n \text{ lorsque } n \leq t < n + 1$$

Lorsque un événement arrive à l'instant t, avec $n \leq t < n + 1$, on considère dans le modèle qu'il est arrivé à l'instant $t = n$. Par conséquent, l'espace du temps est discret et la résolution temporelle pour la modélisation est d'un top d'horloge. Les contraintes temporelles sont spécifiées toujours en termes de top d'horloges.

A chaque événement $a \in \sum_{act}$ on associe un intervalle $[l_a, u_a]$, $l_a \in \mathbb{N}$ et $u_a \in \mathbb{N} \cup \{\infty\}$. Un triplet (a, l_a, u_a) dénote un événement temporel.

Les événements sont classifiés en deux catégories selon la valeur de la borne supérieure de l'intervalle associé.

- Un événement a est appelé *prévu* si $0 \leq u_a < \infty$. L'ensemble des événements prévus est noté $\sum_{spe}$.

- Un événement a est appelé *lointain* si $u_a = \infty$. L'ensemble des événements lointains est noté $\sum_{rem}$.

Par conséquent, l'ensemble des événements $\sum_{act}$ est partitionné en deux sous-ensembles disjoints :

$$\sum_{act} = \sum_{spe} \cup \sum_{rem}.$$

A chaque événement a on associe une temporisation t_a, qui mesure le temps écoulé depuis sa dernière validation.

Pour modéliser les contraintes temporelles dans le modèle du comportement d'un SED on utilise un nouvel automate $\mathcal{A} = (Q, \sum, \delta, q_0, Q_m)$ dérivé de l'automate $\mathcal{A}_{act}$. Cet automate est défini de la façon suivante :

- Q est l'ensemble des états. Chaque état $q \in Q$ mémorise un état logique $q_{act} \in Q$ ainsi que la valeur de la temporisation t_a associée à chaque événement $a \in \sum_{act}$:
$$q = \{q_{act} | \{t_a | a \in \sum_{act}\}\}$$
- $\sum$ est l'ensemble des événements. Le passage du temps est modélisé par l'occurrence d'un événement particulier. Cet événement, noté $tick$, modélise l'occurrence d'un top d'horloge.
$$\sum = \sum_{act} \cup \{tick\}$$
- δ est la fonction de transition. Un événement a peut être exécuté depuis un état q, i.e. $\delta(q, a)!$, si $\delta_{act}(q_{act}, a)!$, et la contrainte temporelle associée à l'occurrence de a est vérifiée.
- q_0 est l'état initial
- Q_m est l'ensemble des états marqués.

Un événement a est considéré valide depuis un état q si $\delta(q_{act}, a)!$. Il devient *éligible*, donc il peut être exécuté, lorsque la contrainte temporelle associée à sa date d'occurrence est vérifiée.

La prise en compte du temps enrichit le modèle étudié, mais en contre partie, augmente sa complexité.

Exemple 1.2. Considérons un SED qui a un seul état logique.

Supposons que les événements qui peuvent se produire dans ce système sont $(a, 1, 1)$ et $(b, 2, 3)$. Le comportement logique du système est modélisé par l'automate $\mathcal{A}_{act} = (Q_{act}, \sum_{act}, \delta_{act}, q_{act,0}, Q_{act,m})$, où :

- $Q_{act} = \{0\}$,
- $\sum_{act} = \{a, b\}$,
- $q_{act,0}\{0\}$,
- $\delta_{act}(0, a) = \delta_{act}(0, b) = 0$,
- $Q_{act,m} = \{0\}$

Cet automate est représenté dans la figure 1.9

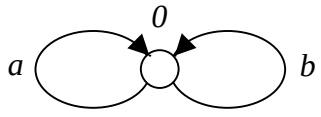


FIG. 1.9 – Automate $\mathcal{A}_{act}$

Le comportement temporel du système est modélisé par l'automate $\mathcal{A} = (Q, \sum, \delta, q_0, Q_m)$ où :

- $Q = \{0\} \times [0, 1] \times [0, 3]$,
- $\sum = \{a, b, tick\}$,

- δ est la fonction de transition,
- $q_0 = \{0\}$,
- $Q_m = \{0\}$.

Cet automate est représenté dans la figure 1.10. Chacun de ses états est caractérisé par une valeur particulière des temporisations associées aux événements. Il faut noter que la prise en compte explicite du passage du temps a engendré l'augmentation du nombre d'états. Ainsi, cet automate a sept états et onze transitions, contrairement à l'automate modélisant le comportement logique du système qui a seulement un état et deux transitions.

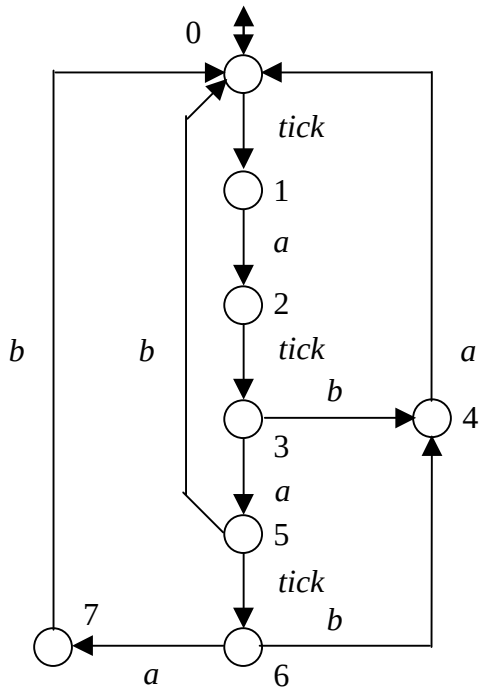


FIG. 1.10 – Automate $\mathcal{A}$

■

De la même manière que pour les SED, certains événements sont contrôlables tandis que les autres ne le sont pas. Ainsi, l'ensemble des événements $\sum$ est partagé en trois sous-ensembles disjoints :

$$\sum = \sum_c \cup \sum_u \cup \{tick\}$$

où :

- $\sum_c$ est l'ensemble des événements contrôlables ;
- $\sum_u$ est l'ensemble des événements incontrôlables.

Un événement contrôlable peut être interdit indéfiniment. Seulement les événements lointains peuvent être contrôlables, $\sum_c \subseteq \sum_{rem}$.

Par opposition, les événements prévus ont des dates d'occurrence au plus tard, au delà ils ne peuvent plus être interdits. Ainsi, les événements prévus sont incontrôlables $\sum_{spe} \subseteq \sum_u$. Certains événements lointains peuvent être incontrôlables par leur nature. Ainsi, l'ensemble des événements incontrôlables est défini par :

$$\sum_u = \sum_{spe} \cup (\sum_{rem} - \sum_c)$$

Une autre catégorie des événements, essentielle dans la commande supervisée des SEDT, est représentée par les événements forçables. Un événement est considéré forçable s'il peut se produire spontanément ou être forcé par un système extérieur tout en respectant la contrainte temporelle associée à sa date d'occurrence. Dans l'approche de Brandin et Wonham, un événement forçable peut préempter l'occurrence de l'événement *tick*. Ainsi, le superviseur peut forcer l'occurrence d'un événement avant que l'horloge atteigne une certaine valeur.

L'ensemble des événements forçables est noté $\sum_{for}$. Il n'y a aucune relation entre l'ensemble des événements forçables et les ensembles des événements contrôlables et incontrôlables. Un événement contrôlable peut ne pas être forçable dans le sens où on peut interdire son occurrence, mais on ne peut pas forcer son exécution. De même, certains événements incontrôlables peuvent être considérés forçables. Intuitivement, le passage de temps ne peut pas être forcé, donc $tick \notin \sum_{for}$.

Remarque 1.1. Dans notre travail de recherche, nous proposons une nouvelle classification des événements. Nous considérons qu'un événement peut être contrôlable ou incontrôlable. Un événement est contrôlable si on peut fixer sa date d'occurrence à l'intérieur d'un intervalle donné. Par contre, un événement est considéré incontrôlable si on ne peut pas agir sur sa date d'arrivée. Ainsi, nous considérons que la notion de forçage d'un événement est équivalente à la notion de contrôlabilité. Cette classification sera détaillée dans le chapitre 4.

■

Soit P un procédé à superviser. On modélise son comportement temporisé avec un automate $\mathcal{P} = (Q, \sum, \delta, q_0, Q_m)$ qui génère un langage $L(\mathcal{P})$. Considérons un mot $w \in L(\mathcal{P})$. Alors il existe un état $q \in Q$ atteint par l'exécution du mot w depuis l'état initial q_0. Les possibilités d'évolution du procédé depuis cet état sont décrites par l'ensemble d'événements éligibles dans l'état q. A chaque mot $w \in L(\mathcal{P})$ on associe un ensemble d'événements éligibles, $Elig_{\mathcal{P}}(w)$, défini par :

$$Elig_{\mathcal{P}}(w) = \{a \in \sum | wa \in L(\mathcal{P})\}$$

Formellement, un superviseur est défini par une fonction

$$S(w) : L(\mathcal{P}) \to 2^{\Sigma}$$

tel que $\forall w \in L((P))$:

$$S(w) \cap Elig_{\mathcal{P}}(w) \neq \phi$$

$$S(w) \supseteq \begin{cases} \sum_u \cup \{tick\} & \text{si} S(w) \cap \sum_{for} = \phi \\ \sum_u & \text{si} S(w) \cap \sum_{for} \neq \phi. \end{cases}$$

De la même façon que dans la théorie de supervision des SED, les événements incontrôlables sont toujours autorisés par le superviseur. Par contre, lorsque parmi les événements éligibles il y a au moins un événement forçable, le superviseur peut forcer son exécution avant l'occurrence d'un nouveau top d'horloge (événement *tick*). Dans cette approche, le superviseur joue aussi un rôle d'un système de commande.

Le comportement en boucle fermée, le concept de contrôlabilité d'un langage ainsi que le calcul du langage suprêmal contrôlable sont traités de la même façon que dans la théorie de Ramagde et Wonham. De plus, on montre que le superviseur trouvé est le plus permissif.

En conclusion, cette approche donne une bonne solution au problème de commande par supervision des SEDT. La matérialisation de l'écoulement du temps à travers un

événement spécifique inclut la dimension temporelle au sein même d'un langage comparable au langage d'un automate non temporisé. Néanmoins, elle a un inconvénient majeur au niveau de la modélisation. Dans la plupart des cas, la nature discrète de l'espace du temps engendre l'explosion combinatoire du nombre d'états du modèle. De plus, l'aspect discret du temps est une approximation dans la modélisation du système.

1.3.2 Commande par supervision en temps continu

Plusieurs approches de commande par supervision ont été proposées pour pallier au problème de l'explosion combinatoire du nombre d'états engendrée par une nature discrète du temps. Ces approches, basées principalement sur l'outil automate temporisé, considèrent que le temps évolue d'une manière continue. Dans cette section nous présentons les approches à temps continu de commande par supervision des SEDT qui sont en relation directe avec nos travaux de recherche.

Commande supervisée par détemporisation

Dans [Gou99] l'auteur propose une approche pour la synthèse de la commande par supervision en s'appuyant sur l'outil automate temporisé [AD94].

Un automate temporisé est un automate fini muni d'un ensemble de variables continues par morceaux appelées horloges. Ces variables mesurent l'écoulement du temps. Lorsque le système séjourne dans un sommet, chaque horloge x a une dynamique continue décrite par l'équation $\dot{x} = 1$. A chaque sommet on associe un prédicat sur la valeur des horloges appelé invariant du sommet. Le système peut séjourner dans un sommet tant que la valeur des horloges vérifie l'invariant associé. Le franchissement d'une transition de l'automate est instantané. A chaque transition on associe une condition de franchissement, appelée garde, et une affectation. Une transition peut être franchie depuis un sommet si sa garde est vérifiée par la valeur des horloges. Les gardes modélisent les contraintes temporelles imposées sur l'évolution du système. L'affectation associée à une transition désigne les horloges mises à zéro par son franchissement.

Considérons, par exemple, l'automate temporisé représenté dans la figure 1.11. Cet automate a deux horloges, x_1 et x_2, qui sont incrémentées dans les sommets. L'invariant du sommet L_0 est $x_2 \leq 1$. La garde associée à la transition de sortie du sommet L_0 est $a \wedge x_2 = 1$. Cette transition peut être franchie depuis L_0 lorsque l'événement a arrive et l'horloge x_2 a la valeur 1. L'affectation associée à cette transition est $x_2 = 0$. Ainsi, le franchissement de cette transition entraîne la mise à zéro de l'horloge x_2. Une présentation formelle de l'outil automate temporisé sera fournie dans le chapitre 2, section 2.2.1.

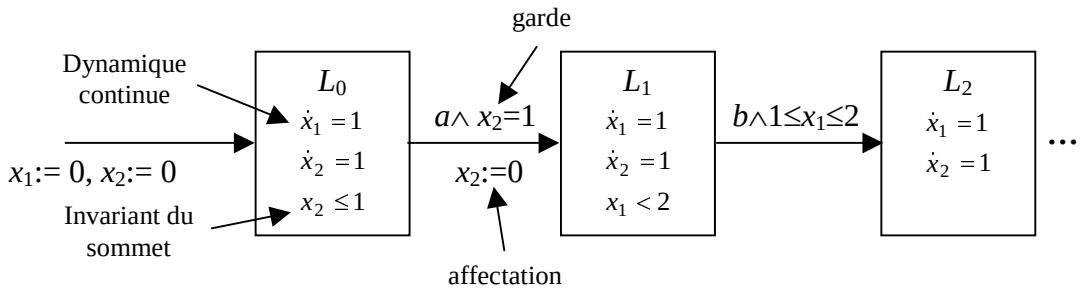


FIG. 1.11 – Automate temporisé

Dans l'approche de commande supervisée par détemporisation, le procédé ainsi que les spécifications sont modélisés par des automates temporisés. La première étape de la synthèse de la commande supervisée consiste à effectuer la composition synchrone de deux automates. Dans le cas des automates temporisés, pour chaque événement commun il faut prendre en compte les contraintes temporelles sur sa date d'occurrence. Ainsi, lors de la composition synchrone des automates, la contrainte temporelle associée à un événement commun est la conjonction des contraintes temporelles qu'il doit vérifier dans chacun des automates temporisés. Cette dépendance temporelle mutuelle peut engendrer l'apparition des incohérences temporelles.

Il y a deux catégories d'incohérences temporelles.

Les incohérences temporelles de la première catégorie apparaissent lorsque on atteint un sommet avec une valeur des horloges telle qu'aucune garde de ses transitions de sortie ne sera jamais vérifiée. La conséquence de ce type d'incohérence temporelle est le blocage dans un sommet de l'automate.

Exemple 1.3. Considérons la partie d'automate temporisé présentée dans la figure 1.12. Par simplification, nous n'avons pas représenté l'évolution des horloges dans les sommets de l'automate. La valeur de l'horloge x_1 à l'entrée dans le sommet L_2 est supérieure à 5, tandis que la garde de l'arc de sortie de la transition de L_2 à L_3 est $x_1 = 3$. Ainsi, cette transition ne pourra jamais être franchie et le système reste bloqué dans le sommet L_2.

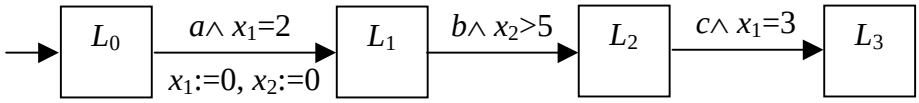


FIG. 1.12 – Premier type d'incohérences temporelles

■

Les incohérences temporelles de la deuxième catégorie apparaissent lorsque la date de franchissement au plus tôt d'une transition de sortie d'un sommet est plus grande que la date de franchissement au plus tard d'une autre transition de sortie du même sommet. La conséquence de ce type d'incohérence temporelle est le fait qu'une transition de sortie du sommet considéré ne sera jamais franchie. Ainsi, elle complique inutilement la représentation de l'automate temporisé.

Exemple 1.4. Considérons la partie d'automate temporisé présentée dans la figure 1.13. La transition de L_1 vers L_3 ne peut jamais être franchie parce que pendant le séjour du système dans L_1, l'horloge x_1 atteint toujours la valeur 2 avant que $x_2 = 6$.

■

La détection des incohérences temporelles est effectuée par une technique de manipulation des intervalles temporels. Cette opération est réalisée à titre préventif avant de commencer la synthèse de la commande par supervision.

La première étape consiste à détemporiser le modèle. L'automate résultant est comparable à un automate non temporisé. Ensuite on applique la théorie de Ramadge et Wonham pour réaliser la synthèse du superviseur.

L'opération de détemporisation d'un automate temporisé est basée sur l'observation que certaines valeurs des horloges, pourtant différentes, engendrent les mêmes possibilités d'évolution depuis le même sommet de l'automate. L'ensemble des valeurs des horloges qui ont cette propriété est appelé *région d'horloges*.

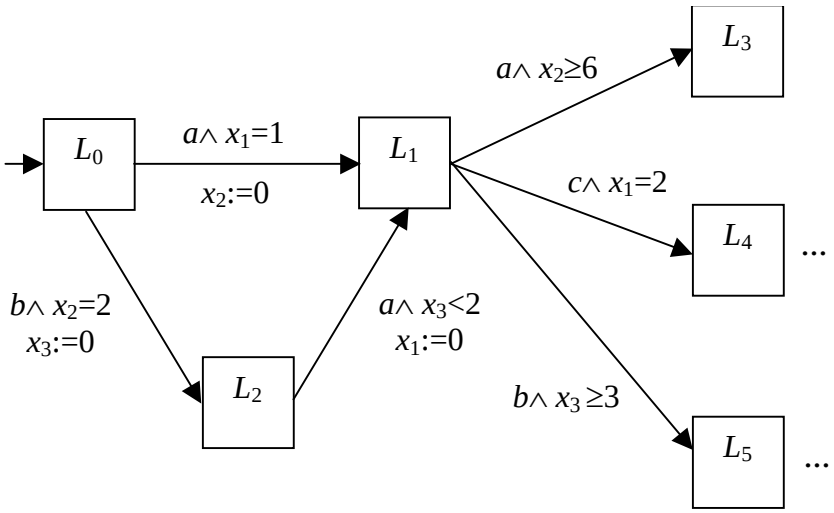


FIG. 1.13 – Deuxième type d'incohérences temporelles

On considère que les contraintes temporelles qui interviennent dans l'évolution d'un automate temporisé sont spécifiées par des nombres entiers. Par conséquent, la partie entière de la valeur d'une horloge détermine si une contrainte temporelle est vérifiée ou non. La connaissance des parties fractionnelles de la valeur des horloges permet de déterminer l'horloge qui accédera en premier à la valeur entière immédiatement supérieure.

Par la suite, nous présentons le concept de région d'horloges à travers un exemple.

Exemple 1.5. Considérons l'automate temporisé $\mathcal{A}$ présenté dans la figure 1.11. Cet automate a deux horloges, x_1 et x_2.

La plus grande constante avec laquelle est comparée l'horloge x_1 est 2. La plus grande constante avec laquelle est comparée l'horloge x_2 est 1. Les régions d'horloges sont présentées dans la figure 1.14. Il y a vingt huit régions :

- 6 points (par exemple [0,1]),
- 14 segments ouverts (par exemple $[(0 < x_1 < 1) \wedge (x_2 = 1)]$ ou $[(0 < x_1 < 1) \wedge (0 < x_2 < 1) \wedge x_1 = x_2]$),
- 8 régions ouvertes (par exemple $[(0 < x_1 < 1) \wedge (x_2 > 1)]$)

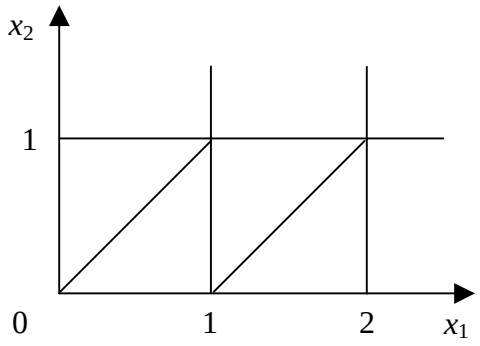


FIG. 1.14 – Régions d'horloges

■

Une région d'horloges α' est appelée *successeur* d'une région d'horloges α si elle est atteinte en laissant le temps évoluer depuis α. Généralement, une région d'horloges peut

avoir plusieurs successeurs.

Une région d'horloges α' est appelée *successeur direct* d'une région d'horloges α si elle est la première région d'horloges rencontrée lorsque le temps commence à évoluer depuis α.

Exemple 1.6. Considérons la région d'horloges $[(0 < x_1 < 1) \wedge (x_2 = 0)]$ présentée dans la figure 1.14. Les successeurs de cette région d'horloges sont :

- $[(0 < x_1 < 1) \wedge (0 < x_2 < 1)]$,
- $[(x_1 = 1) \wedge (0 < x_2 < 1)]$,
- $[(1 < x_1 < 2) \wedge (0 < x_2 < 1)]$,
- $[(1 < x_1 < 2) \wedge (x_2 = 1)]$,
- $[(1 < x_1 < 2) \wedge (1 < x_2)]$,
- $[(x_1 = 2) \wedge (1 < x_2)]$,
- $[(2 < x_1) \wedge (1 < x_2)]$.

Le successeur direct de cette région est $[(0 < x_1 < 1) \wedge (0 < x_2 < 1)]$.

■

La première étape de l'opération de détemporisation consiste à transformer l'automate temporisé qui modélise le procédé et les spécifications en automate de régions. Nous expliquons le principe de cette transformation en considérant l'automate temporisé présenté dans la figure 1.11.

L'automate de régions, noté $\mathcal{A}_R$, obtenu par la transformation de cet automate temporisé est représenté dans la figure 1.15.

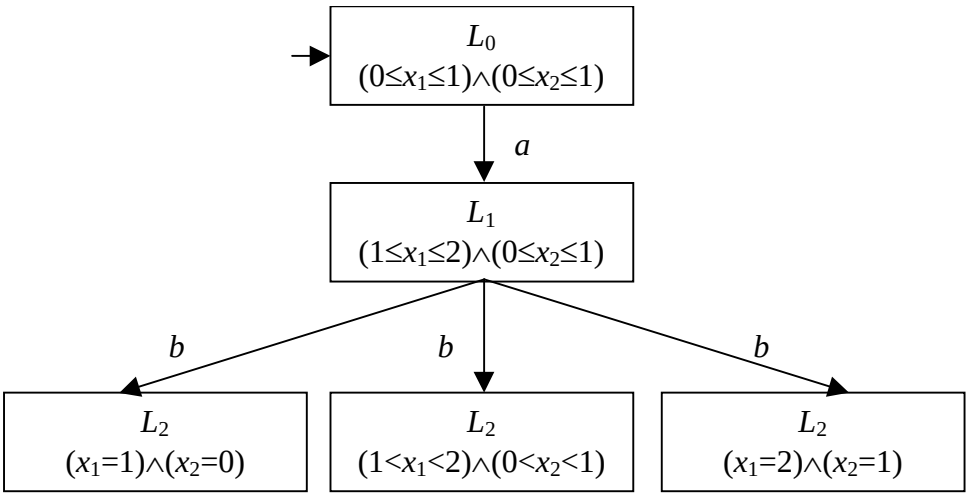


FIG. 1.15 – Automate de régions

L'espace des valeurs des horloges dans un sommet d'un automate de régions est l'union d'un nombre fini de régions d'horloges disjointes [Yov98]. Ainsi, le sommet L_0 contient les régions d'horloge suivantes : $[(x_1 = 0) \wedge (x_2 = 0)]$, $[(0 < x_1 < 1) \wedge (0 < x_2 < 1)]$ et $[(x_1 = 1) \wedge (x_2 = 1)]$. La contrainte temporelle associée à l'occurrence de l'événement a est $x_2 = 1$. Ainsi, cet événement peut être exécuté lorsque la valeur des horloges est dans la région d'horloges $[(x_1 = 1) \wedge (x_2 = 1)]$. Le sommet L_1 contient les régions d'horloge suivantes : $[(x_1 = 1) \wedge (x_2 = 0)]$, $[(1 < x_1 < 2) \wedge (0 < x_2 < 1)]$ et $[(x_1 = 2) \wedge (x_2 = 1)]$. La contrainte temporelle associée à l'occurrence de l'événement b est $1 \leq x_1 \leq 2$. Ainsi, cet événement peut être exécuté lorsque la valeur des horloges est dans chacune des régions d'horloges contenues dans ce sommet.

Pour un automate de régions, l'information temporelle est incluse dans chacun des sommets. Elle n'est plus présente sur les transitions. Par conséquent, un automate de régions peut être considéré comme étant un automate non temporisé. L'automate de régions peut avoir un comportement non déterministe.

La deuxième étape de l'opération de détemporisation consiste à transformer l'automate de régions en automate de τ-régions, noté $\mathcal{A}_\tau$. Son objectif est d'éliminer le non déterminisme. Cette transformation est basée sur l'introduction d'un nouvel événement, noté τ. Cet événement est associé au passage d'une région d'horloges à son successeur direct. Nous illustrons le principe de cette transformation en considérant l'automate de régions $\mathcal{A}_R$, présenté dans la figure 1.15. Le sommet L_1 de cet automate de régions contient trois régions d'horloges. Par conséquent, ce sommet est représenté au niveau de l'automate de τ-régions par trois sommets. La région d'horloge $[(1 < x_1 < 2) \wedge (0 < x_2 < 1)]$ est le successeur direct de la région d'horloges $[(x_1 = 1) \wedge (x_2 = 0)]$. Ainsi, le passage entre les sommets de l'automate de τ-régions correspondants à ces deux régions d'horloges est associé à l'occurrence de l'événement τ. L'automate de τ-régions associé à l'automate de régions $\mathcal{A}_R$ est illustré dans la figure 1.16.

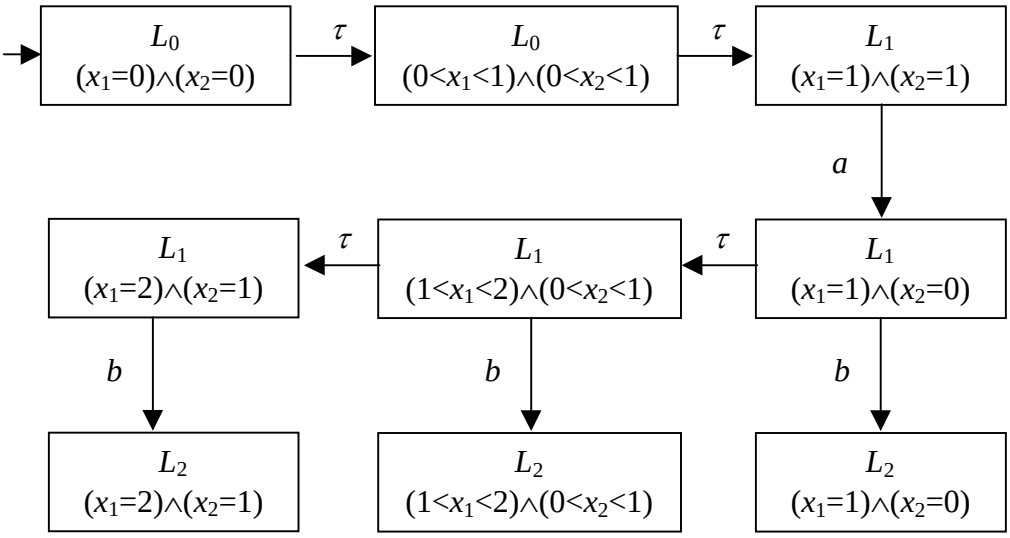


FIG. 1.16 – Automate de τ-régions

L'événement τ modélise d'une certaine façon l'écoulement du temps. L'événement τ ne modélise pas une durée unique. Par opposition, l'événement t, introduit par l'approche de Brandin et Wonham [BW94], est associé a un top d'horloge, donc à une durée constante.

Le modèle utilisé pour la synthèse de la commande par supervision est l'automate de τ-régions. De la même façon que dans l'approche de Brandin et Wonham, on utilise la notion d'événement forçable qui peut préempter l'exécution de l'événement τ. Lorsque τ est en concurrence avec un événement forçable, il est considéré contrôlable et noté avec *tack*. Sinon il est considéré incontrôlable et noté avec *tock*. La synthèse du superviseur est effectuée en appliquant la théorie de Ramadge et Wonham.

L'inconvénient de cette approche réside dans fait que l'ajout d'un événement pour modéliser chaque changement de région d'horloge peut conduire à une explosion du nombre de sommets de l'automate.

Commande par supervision basée sur l'automate temporisé à retards

Dans [AGP+99], les auteurs proposent l'utilisation du modèle réseau de Petri à retards pour la modélisation du processus à commander. La synthèse de la commande par supervision est basée sur l'outil automate temporisé à retards .

L'outil réseau de Petri à retards a la structure discrète du modèle réseau de Petri autonome [DA92] [Mur89], tandis que l'information temporelle est modélise de la même façon que dans le modèle automate temporisé [Yov93] [Oli94]. Un réseau de Petri à retards [AGP+99] a des horloges et des contraintes temporelles sur le franchissement des transitions. Chaque transition modélise l'occurrence d'un événement. Ainsi, les transitions sont classifiées en contrôlables ou incontrôlables selon la nature de l'événement modélisé.

De même que dans le modèle automate temporisé, à chaque transition d'un réseau de Petri à retards on associe une condition de franchissement et une affectation.

La contrainte temporelle associée à la date d'occurrence d'un événement est modélisée par la garde associée à la transition correspondante. Une transition est validée si chacune de ses places d'entrée contient au moins une marque. Elle peut être franchie seulement si sa garde est vérifiée par la valeur des horloges.

L'affectation associée à une transition modélise la mise à zéro de certaines horloges par son franchissement. Une horloge mise à zéro par le franchissement d'une transition peut être testée par al garde de n'importe quelle transition du réseau.

Nous illustrons le modèle RdP à retards à travers un exemple.

Exemple 1.7. Considérons le RdP à retards présenté dans la figure 1.17.

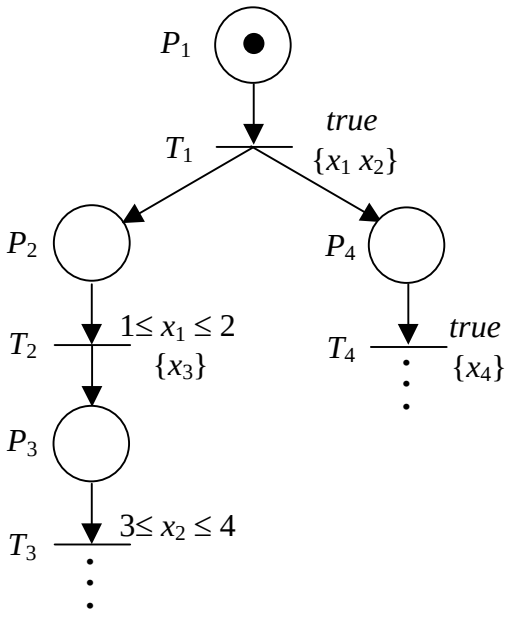


FIG. 1.17 – Réseau de Petri à retards

Initialement, seulement la place P_1 est marquée. Ce marquage du réseau permet al validation de la transition T_1. La garde associée à cette transition est *true*, donc il n'y a aucune contrainte temporelle sur son franchissement. L'affectation associée à T_1 est $\{x_1, x_2\}$. Par conséquent, ces horloges sont mises à zéro par son franchissement.

L'horloge x_1 est testée par la garde associée à al transition T_2. Cette transition peut être franchie si elle est validée par la marquage du RdP sous-jacent et 1 à 2 u.t. se sont écoulées depuis le franchissement du T_1.

Par contre, l'horloge x_2 est testé par la garde de la transition T_4. Celle-ci peut être franchie si elle est validée par le marquage du réseau et si 3 à 4 u.t. se sont écoulées depuis le franchissement de T_1.

■

Par conséquent, un réseau de Petri à retards évolue de la même façon qu'un automate temporisé. Ce qu'il apporte de plus par rapport à l'automate temporisé est la possibilité de représenter explicitement certains mécanismes comme le parallélisme, la synchronisation et le partage des ressources.

L'outil réseau de Petri à retards est utilisé seulement pour la modélisation du procédé, sans prendre en compte les spécifications imposées sur son fonctionnement.

L'étape suivante consiste à modéliser le comportement du réseau de Petri à retards par un automate temporisé à retards. Cet outil est dérivé du modèle automate temporisé en remplaçant les invariants des sommets par des conditions de franchissement au plus tard associées aux transitions.

La spécification imposée sur le fonctionnement du procédé est exprimée par une propriété qui doit être satisfaite par les états de l'automate temporisé à retards correspondant au procédé.

Soit un automate temporisé à retards $\mathcal{A}$ et une propriété S. L'objectif de la synthèse d'un superviseur est de construire un automate à retards $\mathcal{A}_S$ tel que tous ses états vérifient la propriété S.

Soit Q le sous-ensemble des états de l'automate $\mathcal{A}$ qui satisfont la propriété S. La procédure de synthèse, est basée sur l'approche présenté dans [MPS95]. Cette méthode permet de déterminer l'ensemble $\pi(Q)$ des états à partir desquels on peut atteindre n'importe quel état $q \in Q$ par le franchissement d'une transition contrôlable. On enlève de l'ensemble $\pi(Q)$ les états à partir desquels on peut évoluer, par le franchissement d'une transition incontrôlable, vers un état qui n'appartient pas au sous-ensemble Q.

La procédure de calcul du superviseur est itérative. Elle est initialisée avec $Q[0] = Q$. A chaque itération i on calcule l'ensemble $\pi(Q[i])$ et on réactualise $Q[i+1] = Q[i] \cap \pi(i)$. L'algorithme s'arrête lorsque $Q[i+1] = Q[i]$. L'ensemble obtenu est noté avec Q^*. L'automate $\mathcal{A}_S$ obtenu par cette procédure de synthèse de la supervision a la même structure discrète que l'automate $\mathcal{A}$ sauf pour les gardes des arcs contrôlables qui sont modifiées telles que seulement les états $q \in Q^*$ soient atteignables.

Cette approche permet de déterminer toutes les séquences de transitions contrôlables tel que le système n'évolue pas vers des états qui ne respectent pas la propriété S.

L'outil réseaux de Petri à retards permet de représenter d'une façon explicite des comportements des SEDT comprenant le parallélisme, la synchronisation et le partage des ressources. Cependant, l'indépendance entre la structure graphique et la valuation des horloges rend difficile l'interprétation du modèle. Un autre inconvénient de l'outil réseau de Petri à retard est que chacune de ses places peut avoir au plus une marque, i.e. il est sauf. Par conséquent, il n'est pas approprié pour modéliser le comportement des systèmes de production.

Commande par supervision basée sur l'automate à temps continu

L'approche présentée dans [Kou99] propose l'utilisation du modèle réseau de Petri à arcs temporels pour la modélisation du procédé et des spécifications imposées. La synthèse de la commande par supervision s'appuie sur l'outil automate à temps continu.

L'outil réseau de Petri (RdP) à arcs temporels est dérivé du modèle réseau de Petri autonome en lui associant des intervalles temporels aux arcs de sortie des places. Ces intervalles modélisent les contraintes temporelles qui interviennent dans le fonctionnement du système. L'occurrence des événements dans le procédé est modélisée par le franchissement des transitions. Les contraintes temporelles sur la date d'occurrence d'un événement sont représentées par les intervalles temporels associés aux arcs d'entrée dans la transition correspondante. Une transition d'un RdP à arcs temporels peut être contrôlable ou incontrôlable selon la nature de l'événement modélisé.

Nous présentons le principe de fonctionnement du modèle RdP à arcs temporels à travers un exemple.

Exemple 1.8. Considérons le RdP à arcs temporels illustré dans la figure 1.18.

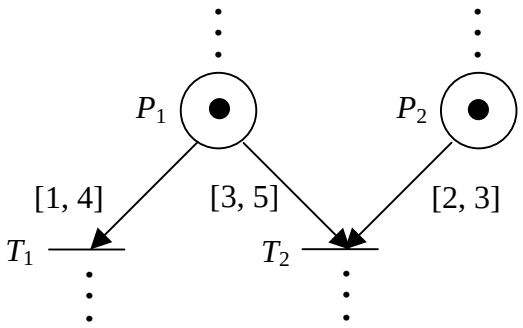


FIG. 1.18 – Réseau de Petri à arcs temporels

L'intervalle de franchissement associé à l'arc $P_1 \rightarrow T_1$ est $[1, 4]$. Dans ce cas, une marque présente dans la place P_1 peut participer à la validation (et au franchissement) de la transition T_1 lorsque 1 à 4 u.t. se soit écoulées depuis l'instant de son arrivée dans P_1. Ainsi, on dit que la marque est indisponible pour la validation de T_1 pendant une u.t. Ensuite elle devient disponible et conserve cette propriété jusqu'à ce que 4 u.t. se soit écoulées depuis son arrivée dans P_1. Une fois cette période dépassée, la marque ne peut plus jamais participer à la validation de T_1.

La même marque est indisponible pendant 3 u.t. pour la validation de la transition T_2. Elle peut être prise en compte pour la validation de cette transition lorsqu'elle y est séjourné 3 à 5 u.t. dans P_1.

Le modèle du comportement désiré du procédé est obtenu en effectuant le produit synchrone des RdP à arcs temporels qui modélisent le procédé à commander et les spécifications imposées par le cahier de charges.

Un état du procédé est défini par le marquage des places du RdP à arcs temporels qui modélise son comportement. Une évolution qui mène le procédé dans un état qui ne respecte pas les spécifications imposées sur son fonctionnement, est appelée non désirée. Les états atteints par des évolutions non désirées sont appelés interdits.

L'outil RdP à arcs temporels hérite la capacité de modélisation des modèles RdP. Par contre, il est moins approprié pour l'analyse. Généralement, l'analyse d'un système modélisé par l'outil réseau de Petri nécessite la construction du graphe de marquage qui modélise son comportement.

Cette approche utilise l'outil automate à temps continu pour la synthèse de la commande par supervision. Par conséquent, l'étape suivante consiste à construire l'automate à temps continu qui modélise le comportement du réseau de Petri à arcs temporels.

Chaque état du RdP à arcs temporels est modélisé par un sommet de l'automate à temps continu. Ainsi, les états interdits sont modélisés par des sommets interdits.

Chaque franchissement d'une transition du RdP est modélisé par le franchissement d'une transition de l'automate. Les transitions d'un automate à temps continu sont contrôlables ou incontrôlables selon la nature de la transition correspondante dans le modèle RdP à arcs temporels. Les contraintes temporelles sur la date de franchissement des transitions d'un RdP à arcs temporels sont modélisées par des gardes associées aux transitions de l'automate.

La synthèse de la commande par supervision consiste à analyser un par un tous les sommets interdits et à restreindre les gardes des transitions controlâbles telles que ces sommets ne soient plus jamais atteignables. Cette méthode est basée sur une technique de calcul du temps minimal et maximal de séjour du système dans un sommet de l'automate.

Nous présentons le principe de la méthode de synthèse en considérant la partie d'automate à temps continu présentée dans la figure 1.19.

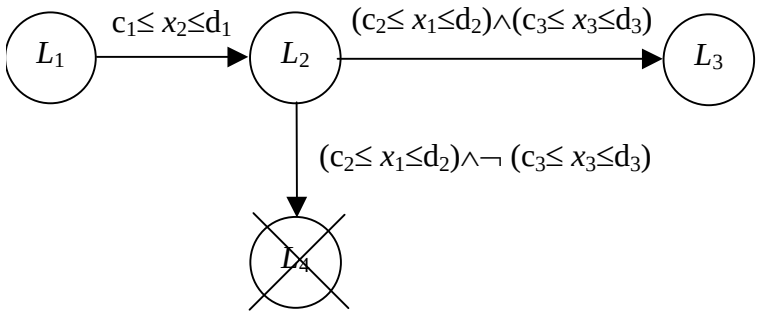


FIG. 1.19 – Partie d'automate à temps continu

Le sommet interdit L_4 est atteint à partir du sommet L_2. L'objectif de la synthèse de la commande par supervision est de garantir qu'à partir du sommet L_2, la transition vers L_3 est toujours franchie avant la transition vers le sommet interdit L_4. Par conséquent, on cherche des nouvelles valeurs pour les gardes des transitions contrôlables d'entrée et de sortie de L_2, i.e. $T_{1,2}$ et $T_{2,3}$, qui garantissent que le temps maximal de séjour dans L_2 avant d'évoluer vers L_3 soit plus petit que le temps minimal de séjour dans L_2 avant d'évoluer vers L_4.

La technique de synthèse proposée par cette approche est basée sur deux procédures.

La première consiste à modifier la garde de la transition de sortie de L_2 vers L_3 telle que cette transition soit franchie toujours avant la transition vers le sommet interdit. Cette procédure, appelée *contrôle aval* peut être appliquée seulement si la transition de L_2 vers L_3 est contrôlable.

La deuxième procédure consiste à modifier la garde de la transition d'entrée dans L_2 telle que la valeur des horloges dans ce sommet ne permette pas de franchir la transition vers L_4. Cette procédure, appelée *contrôle amont* peut être appliquée seulement pour les transitions d'entrée qui sont contrôlables. Lorsqu'aucune de ces procédures ne permet de trouver une solution, le sommet L_2 devient un sommet interdit. Par conséquent, cette approche cherche une solution locale pour éviter l'évolution du système vers un sommet interdit.

En conclusion, cette approche de synthèse de la commande par supervision associe la capacité de modélisation de l'outil réseau de Petri à arcs temporels avec la puissance d'analyse du modèle automate à temps continu. D'un côté, cette démarche permet d'obtenir un modèle clair et concis du procédé étudié. D'un autre côté, on évite l'apparition

des incohérences temporelles mises en évidence dans [Gou99].

Par contre, la procédure de synthèse proposée s'adresse à une classe particulière d'automates à temps continu où les horloges non couplées, i.e. il n'y a aucune relation entre la valeur des différentes horloges. De même, les gardes des transitions de l'automate sont représentées par des expressions logiques des contraintes sur la valeur de plusieurs horloges. Ainsi, la synthèse devient difficile pour des systèmes de grande taille.

Commande robuste d'un atelier de traîtement de surfaces

Dans [Che99] l'auteur propose une approche pour la synthèse d'une conduite robuste pour des systèmes de production sans stock et sans attente. Le système de production étudié dans le cadre de ce travail est un atelier de traitement de surfaces. Le pilotage d'un tel système est un problème difficile. D'un côté, les opérations ont une durée minimale et une durée maximale. D'un autre côté, elles doivent se soumettre à des contraintes de synchronisation très fortes. De plus, dans cette étude on suppose que le fonctionnement du système est sujet à des perturbations qui peuvent modifier les contraintes temporelles imposées sur l'exécution de certaines tâches.

La spécification imposée sur le fonctionnement de l'atelier de traitement de surfaces est exprimée par un séquencement donné pour les opérations participant au processus de production.

Le comportement du procédé avec sa spécification est représenté par un modèle réseau de Petri P-temporel [Kha97]. Cet outil est une extension du modèle réseau de Petri autonome qui fournit une représentation claire et concise des contraintes temporelles de synchronisation.

L'objectif de la synthèse de la commande est de déterminer les contraintes temporelles associées à l'exécution des tâches tel que le séquencement des opérations imposé pour le fonctionnement du procédé soit respecté.

La robustesse face aux perturbations est assurée par deux méthodes. Pour les perturbations de faible amplitude, on recalcule la commande. Ce calcul est basé sur la résolution d'un programme linéaire déduit du modèle réseau de Petri P-temporel du comportement désiré du procédé. Par contre, pour les perturbations fortes, on choisit un autre séquencement pour les opérations du procédé./par Nous allons montrer dans le chapitre 2 que le modèle réseau de Petri P-temporel n'est pas propre dans le sens que le marquage des places peut évoluer même lorsqu'il n'y a pas de franchissement d'une transition. Cette caractéristique rend difficile l'interpretation d'un modèle réseau de Petri P-temporel.

1.4 Conclusion

La synthèse de la supervision pour les SED est basée sur la théorie de Ramadge et Wonham. Cette théorie ne peut pas être appliquée pour la commande des SED parce que le rôle du superviseur est limité à autoriser ou interdire l'occurrence de certains événements. Dans la littérature, ce problème a été résolu par l'introduction d'une nouvelle catégorie d'événements. Il s'agit des événements forçables. Cette catégorie désigne les événements qui peuvent se produire de façon instantanée ou être forcés par un système extérieur. Dans ce contexte, le superviseur a aussi un rôle de système de commande, dans la mesure où il peut forcer l'occurrence de certains événements. Plusieurs travaux ont été effectués et des méthodes pour la synthèse des lois de commande efficaces ont été proposées. Cependant ces lois de commande sont "contraignantes" dans le sens où on considère qu'un événement peut avoir lieu n'importe quand dans l'intervalle $[0, \infty)$.

Dans la plupart des cas, on dispose d'une information supplémentaire sur la date d'occurrence des événements. La prise en compte de cette information a fait l'objet de travaux sur la synthèse de la commande par supervision des SEDT, pour élaborer des lois de commande moins contraignantes. L'étude de ces approches nous a permis de faire la classification suivante :

1. Les approches à temps discret modélisent le passage du temps par l'occurrence d'un événement. Ces approches sont basées sur la théorie de Ramadge et Wonham et fournissent des lois de commandes efficaces. Par contre, elles présentent l'inconvénient de fournir des modèles d'analyse de taille importante.
2. Les approches à temps continu permettent de contourner le problème de l'explosion du nombre d'états et de l'approximation engendrée par la discrétisation du temps. Plusieurs approches à temps continu ont été étudiées. Nous les classifions selon les outils employés pour la modélisation du procédé à commander et la synthèse de la commande par supervision.
   - L'approche de commande par détemporisation utilise l'outil automate temporisé pour la modélisation du procédé à commander ainsi que pour la synthèse de la commande par supervision. Cette approche hérite des inconvénients de modélisation spécifiques aux automates.
   - Une autre approche, proposé dans [AGP+99], utilise l'outil réseau de Petri à retards pour la modélisation du procédé à commander. La synthèse de la commande est effectuée en s'appuyant sur l'outil automate temporisé à retards. La spécification imposée sur le fonctionnement du système est exprimée par une propriété sur les états de l'automate temporisé. Cette approche est peu adaptée pour la modélisation et la synthèse de la commande pour des systèmes de production.
   - L'approche introduite dans [Kou99] utilise l'outil réseau de Petri à arcs temporels pour la modélisation du procédé et de la spécification imposée. La synthèse de la commande est basée sur l'outil automate à temps continu. Cette approche est très appropriée pour la modélisation et la commande des systèmes de production. Cependant, la méthode proposée par cette approche fournit une solution locale. Elle s'adresse au cas particulier où les variables qui modélisent l'écoulement du temps sont non couplées.
   - Une autre approche étudiée concerne la commande robuste d'un atelier de traîtement de surfaces. Cette approche utilise le modèle réseau de Petri P-temporel pour la modélisation du procédé avec ses spécifications. La synthèse de la commande est effectuée en résolvant un problème de programmation linéaire déduit du modèle réseau de Petri du comportement désiré du procédé.

L'approche que nous proposons pour la synthèse de la commande par supervision est basée sur les outils réseau de Petri T-temporel et automate temporisé. Ces outils, ainsi que les arguments qui justifient nos choix seront présentés dans le chapitre 2. Le procédé à commander, ainsi que les spécifications imposées sur son fonctionnement sont modélisés par des réseaux de Petri T-temporels. Par contre, la synthèse de la commande s'appuie sur l'outil automate temporisé.

De plus, nous proposons une définition unifiée pour la contrôlabilité des événements. Dans notre acception un événement est contrôlable si on peut fixer sa date d'occurrence dans un intervalle donné. A l'opposé, un événement est incontrôlable si on ne peut pas agir sur sa date d'occurrence. Cette classification sera détaillée dans le chapitre 4.

Chapitre 2

Outils de modélisation et d'analyse des SEDT

Dans ce chapitre, nous présentons les outils de modélisation et d'analyse des SEDT auxquels nous nous sommes intéressé dans nos travaux de recherche : les réseaux de Petri et les automates temporisés. Dans un premier temps, nous concentrons notre attention sur les outils réseau de Petri P-temporels et réseau de Petri T-temporels. Ces modèles héritent la puissance de modélisation de l'outil réseau de Petri autonome. De plus ils permettent de représenter la plupart des contraintes temporelles qui interviennent dans le fonctionnement d'un SEDT.

Ensuite, nous présentons le modèle automate temporisé. Nous insistons sur les résultats concernant l'analyse d'atteignabilité des états que nous allons utiliser dans l'approche que nous proposons pour la synthèse de la commande des SEDT.

2.1 Le modèle réseau de Petri

Le modèle réseau de Petri (RdP) est un outil de modélisation et d'analyse très utilisé pour l'étude des systèmes à événements discrets (systèmes de production, protocoles de communications, etc.)[DA92] [DH94] [Dav91]. Sa puissance de modélisation résulte de sa capacité à représenter d'une manière intuitive et naturelle les principaux mécanismes des systèmes à événements discrets : parallélisme, synchronisation et partage des ressources. Néanmoins, le modèle RdP autonome n'offre que des possibilités d'analyse qualitative. Certaines extensions à ce modèle ont été proposées pour prendre en compte explicitement le passage du temps. Parmi ces extensions, nous avons retenu les modèles RdP P-temporel [Kha97] et RdP T-temporel [MF76] [BD91]. Notre choix est motivée par leur capacité à modéliser la plupart des contraintes temporelles intervenant dans le fonctionnement d'un système.

Par la suite, avant de focaliser notre attention sur les outils RdP P-temporel et RdP T-temporel, nous faisons une brève présentation du modèle RdP autonome.

2.1.1 Le modèle RdP autonome

Un RdP autonome est un graphe biparti dont les nœuds sont des places et des transitions reliées par des arcs. L'ensemble des places est fini et non nul. De même, l'ensemble des transitions est fini et non nul. Les arcs sont orientés. Chaque arc relie une place à une transition ou une transition à une place. Des poids (nombres entiers strictement positifs) peuvent être attribués aux arcs.

Avant de donner la définition formelle du modèle RdP autonome, nous illustrons ces concepts de base à travers un exemple.

Exemple 2.1. Considérons le système représenté dans la figure 2.1.

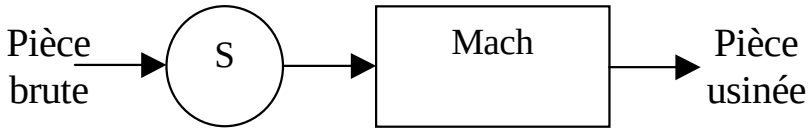


FIG. 2.1 – Machine avec stock d'entrée

Ce système est composé d'un stock S, de capacité finie, et d'une machine $Mach$ qui traîte une par une les pièces déposées dans le stock. On suppose que le stock a une capacité de 2 pièces. De même, tant que le stock n'est pas plein, des pièces sont déposées une par une. Le fonctionnement du système est modélisé par le RdP autonome illustré dans la figure 2.2.

Le stock S est modélisé par la place P_2 et sa capacité par la place P_3. Chaque place contient un nombre entier positif ou nul de marques. Les marques présentes dans la place P_2 correspondent au nombre des pièces dans le stock. Par contre, les marques présentes dans la place P_3 correspondent au nombre de places disponibles dans le stock. Le nombre de marques contenues dans la place P_i est m_i. Pour l'exemple considéré, $m_1 = 1$, $m_2 = 0$, $m_3 = 2$ et $m_4 = 1$. Le marquage d'un RdP, noté M, est défini par le vecteur des marquages des places. Le marquage du RdP présenté dans la figure 2.2 est $M = [1, 0, 2, 1]$.

L'alimentation du stock est modélisée par le franchissement de la transition T_1. Cette transition est considérée validée et peut être franchie si et seulement s'il y a au moins une marque dans chacune de ses places d'entrée, i.e. P_1 et P_3. Le franchissement de T_1

consiste à enlever une marque de chacune de ses places d'entrée et à ajouter une marque dans chacune de ces places de sortie, i.e. P_1 et P_2. La présence d'une seule marque dans la place P_1 modélise le fait qu'une seule pièce à la fois peut être déposée dans le stock. La transition T_2 modélise le traîtement effectué par la machine.

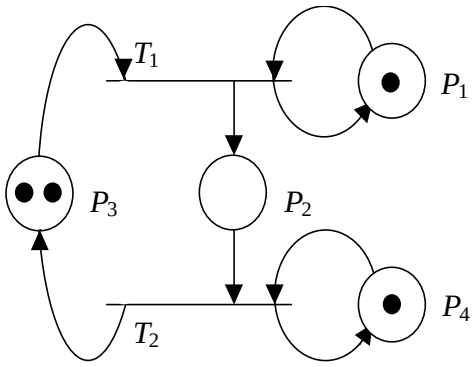


FIG. 2.2 – Réseau de Petri autonome

■

Formellement, un RdP autonome est défini comme suit [DA92] :

Définition 2.1. Un RdP autonome est un quintuplet R=(P, T, $Pré$, $Post$, M_0), où :

- $P = \{P_1,\ P_2, \ldots P_n\}$ est l'ensemble fini des places ;
- $T = \{T_1,\ T_2, \ldots T_m\}$ est l'ensemble fini des transitions, $P \cap T = \phi$;
- $Pré$ est l'application d'incidence avant, telle que :

  $Pré : P \times T \rightarrow \mathbb{N}$,
  $(P_i, T_j) \rightarrow Pré(P_i, T_j)$= le poids de l'arc reliant la place P_i à la transition T_j ;

- $Post$ est l'application d'incidence arrière, telle que :

  $Post : P \times T \rightarrow \mathbb{N}$,
  $(P_i, T_j) \rightarrow Post(P_i, T_j)$= le poids de l'arc reliant la transition T_j à la place P_i ;

- M_0 est le marquage initial.

■

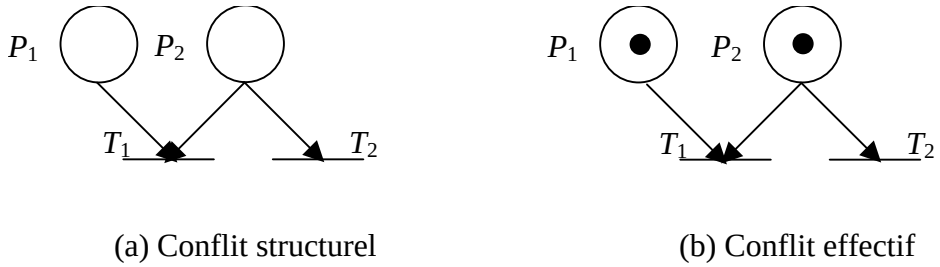
L'état d'un RdP autonome est défini par le marquage des places.

Le comportement d'un RdP autonome est déterminé par l'évolution du marquage engendrée par les franchissements des transitions. Une transition T_j d'un RdP autonome peut être franchie si chacune de ses places d'entrée P_i contient au moins un nombre de marques égal à $Pré(P_i,\ T_j)$. Dans ce cas on dit qu'elle est validée par le marquage du réseau. Le franchissement d'une transition T_j a pour conséquence le retrait d'un nombre de $Pré(P_i,\ T_j)$ marques de chacune de ses places d'entrée P_i et l'ajout d'un nombre de $Post(P_k,\ T_j)$ marques dans chacune de ses places de sortie P_k.

Lorsque deux ou plusieurs transitions ont une place d'entrée commune, on dit qu'il y a un conflit structurel. Un exemple de conflit structurel est illustré dans la figure 2.3.a. Un conflit structurel devient effectif lorsque le marquage d'une place ne permet pas de franchir toutes ses transitions de sortie validées. Par exemple, dans la figure 2.3.b, les transitions T_1 et T_2 sont toutes les deux validées par le marquage de la place P_1, mais seulement une

peut être franchie. Considérons qu'on franchit la transition T_1. Le franchissement de cette transition a pour conséquence le retrait d'une marque de la place P_1. Ainsi, la transition T_2 n'est plus validée, donc elle ne peut plus être franchie.

Un conflit effectif introduit un non déterminisme concernant les transitions franchies dans un RdP. Ce non déterminisme peut être éliminé en choisissant une politique de résolution de conflits appropriée. Considérons le conflit effectif présenté dans la figure 2.3.b. Un exemple de politique de résolution pour ce conflit est de donner la priorité au franchissement de la transition T_1 par rapport à la transition T_2.



(a) Conflit structurel (b) Conflit effectif

FIG. 2.3 – Conflit dans le modèle RdP autonome

La méthode principale utilisée pour déterminer les propriétés d'un RdP est basée sur l'établissement du graphe de marquage qui modélise son comportement. Les nœuds d'un graphe de marquage correspondent aux marquages accessibles, tandis que ses arcs modélisent les franchissements des transitions faisant passer d'un marquage à un autre.

Considérons le RdP autonome présenté dans la figure 2.2, avec le marquage initial M_0=[1, 0, 2, 1]. Son graphe de marquage est présenté dans la figure 2.4.

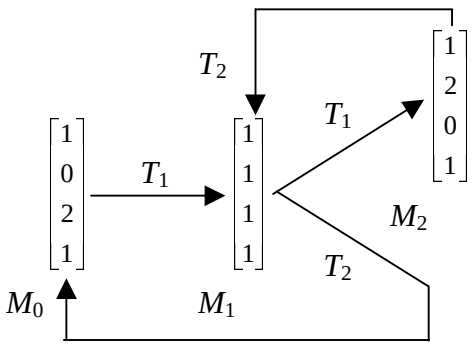


FIG. 2.4 – Graphe de marquage

A partir de M_0, seulement la transition T_1 peut être franchie. On atteint alors le marquage M_1=[1, 1, 1, 1]. Pour ce marquage, les transitions T_1 et T_2 sont toutes les deux validées. Le marquage atteint par le franchissement de T_1 est M_2=[1, 2, 0, 1]. Par contre, si l'on franchit T_2, on retrouve le marquage initial M_0. A partir du marquage M_2, seule la transition T_2 est validée. Son franchissement permet de retrouver le marquage M_1.

Un RdP autonome évolue indépendamment des événements extérieurs. En effet, le franchissement des transitions d'un RdP autonome ne dépend que du marquage. Par conséquent, un RdP autonome permet de représenter uniquement le séquencement logique des événements qui interviennent dans le fonctionnement du système. Ces réseaux forment la classe des RdP de base.

Des extensions du modèle RdP autonome ont été définies pour représenter non seulement ce qui se passe, mais aussi quand ça se passe. Ces extensions permettent une représentation quantitative des comportements modélisés. Dans notre travail nous nous sommes intéressés plus particulièrement à deux modèles, le RdP P-temporel et le RdP T-temporel, que nous présentons par la suite.

2.1.2 Le modèle RdP P-temporel

Le modèle RdP P-temporel a été introduit dans [Kha97] pour modéliser le comportement des systèmes de production dont le fonctionnement est soumis à des contraintes de synchronisation très fortes. Cet outil est dérivé du modèle RdP autonome en lui associant un intervalle de temps $[a_i, b_i]$ à chaque place P_i. Généralement, lorsqu'une marque arrive dans une place, elle ne peut pas participer tout de suite à la validation d'une transition. Dans ce cas, on dit que la marque est indisponible. Une marque dans la place P_i participe à la validation de ses transitions de sortie si elle est restée au moins durant le temps a_i dans cette place. Elle doit quitter la place P_i par le franchissement d'une de ses transitions de sortie au plus tard lorsque sa durée de séjour atteint la borne maximale b_i de l'intervalle de temps associé. Une fois cette valeur dépassée, la marque perd sa capacité à valider des transitions. On dit qu'elle devient morte et elle ne peut plus jamais participer à la validation d'une transition.

Nous illustrons le principe de fonctionnement d'un RdP P-temporel à travers un exemple de modélisation d'une cuve de traitement chimique [Che99].

Exemple 2.2. Considérons une partie d'un système de production constituée d'une cuve de traitement chimique et d'un système de transport représenté par un robot. Cette partie du système de production est illustrée dans la figure 2.5.

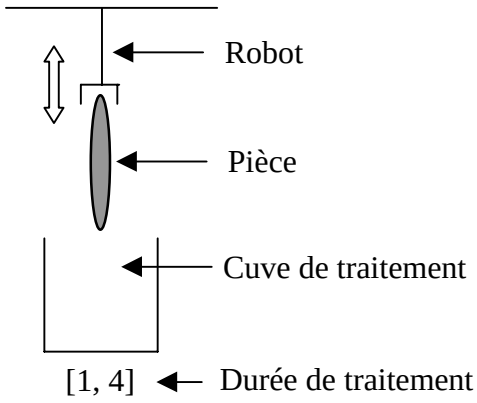


FIG. 2.5 – Partie d'un système de production

La durée de traitement d'une pièce dans la cuve doit être de 1 à 4 unités de temps (u.t.). Ainsi, on considère qu'avant 1 u.t., le traitement n'est pas encore achevé. Par contre, lorsque la durée d'immersion dans la cuve dépasse 4 u.t., la pièce est considérée comme rebut. Le rôle du robot est de transporter les pièces brutes dans la cuve et de retirer les pièces traitées.

Pour simplicité, nous considérons le traitement d'une seule pièce. Le fonctionnement de ce procédé est modélisé par le RdP P-temporel représenté dans la figure 2.6.

La transition T_1 modélise l'arrivée du robot au dessus de la cuve de traitement. La présence d'une marque dans la place P_2 modélise l'état de disponibilité du robot au dessus

de la cuve de traitement. On suppose qu'il n'y a aucune contrainte sur la durée d'attente du robot. Par conséquent, l'intervalle de temps associé à la place P_2 est $[0, \infty)$.

Le traitement d'une pièce dans la cuve est modélisé par la présence d'une marque dans la place P_1. La contrainte temporelle imposée sur la durée de traitement est modélisée par l'intervalle de temps $[1, 4]$ associé à cette place.

Le traitement correct d'une pièce est modélisé par le franchissement de la transition T_2. Une marque dans la place P_1 devient donc disponible pour la validation de T_2 après y avoir séjourné 1 u.t. Elle doit quitter cette place par le franchissement de T_2 au plus tard lorsque la durée de son séjour devient 4 u.t. S'il y a une marque dans P_2 au plus tard à 4 u.t. depuis le dépôt d'une marque dans P_1, la transition T_2 peut être franchie. Cette évolution correspond à un traitement correct de la pièce. Par contre, s'il n'y a pas de marque dans P_2 avant la mort de la marque dans P_1 (i.e. au plus tard à 4 u.t. depuis son dépôt dans P_1), la transition T_2 n'est pas validée, donc elle ne peut pas être franchie. Cette évolution correspond à l'arrivée du robot en retard par rapport à la durée maximale d'immersion d'une pièce dans la cuve. La mort de la marque dans P_1 modélise la transformation de la pièce considérée en rebut.

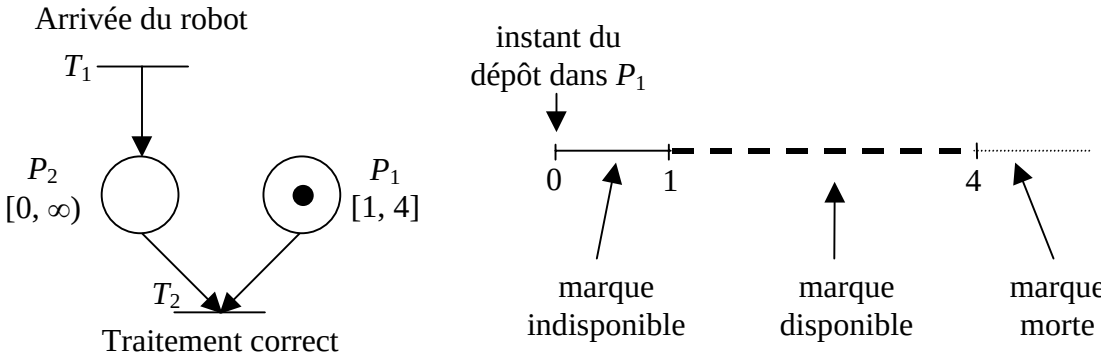


FIG. 2.6 – RdP P-temporel

■

Formellement, un RdP P-temporel est défini comme suit [Kha97] :

Définition 2.2. Un RdP P-temporel est un 6-uplet $< P, T, Pré, Post, M_0, Is >$, tel que :

- $P = \{P_1, P_2, \ldots P_n\}$ est l'ensemble fini des places ;
- $T = \{T_1, T_2, \ldots T_m\}$ est l'ensemble fini des transitions, $P \cap T = \phi$;
- $Pré$ est l'application d'incidence avant ;
- $Post$ est l'application d'incidence arrière ;
- M_0 est le marquage initial ;
- $Is : P \rightarrow \mathbb{Q} \times (\mathbb{Q} \cup \infty)$ est une fonction qui associe un intervalle de temps à chaque place. $\mathbb{Q}$ représente l'ensemble des nombres rationnels positifs.

$$Is(P_i) = [a_i^s, b_i^s]$$
$$a_i^s \leq b_i^s$$
$$0 \leq a_i^s < \infty$$
$$0 \leq b_i^s \leq \infty$$

où a_i^s et b_i^s sont des nombres rationnels positifs.

■

Une marque dans la place P_i participe à la validation des transitions de sortie de P_i si elle y est restée au moins pendant une durée a_i^s. Par contre, elle doit quitter P_i au plus tard lorsque la durée de son séjour atteint la valeur b_i^s.

L'état d'un RdP P-temporel est défini par le couple $S=(M, I)$, où :

- M est le marquage des places du RdP autonome sous-jacent ;
- I est une application qui associe un intervalle $[a_i^j, b_i^j]$ à chaque marque j dans la place P_i. Cet intervalle est défini par rapport à la date d'arrivée de la marque j dans P_i. Il mémorise les dates de disponibilité de la marque j pour participer à la validation des transitions de sortie de P_i.

L'évolution d'un RdP P-temporel est déterminée par de deux types d'événements : la mort des marques dans des places et le franchissement des transitions.

Une transition T_j d'un RdP P-temporel est validée et peut être franchie si chacune de ses places d'entrée P_i contient au moins $Pré(P_i, T_j)$ marques disponibles. Considérons le RdP P-temporel illustré dans la figure 2.6. La transition T_2 est validée s'il y a une marque dans la place P_1 qui y a séjourné pendant une durée de 1 à 4 u.t. et s'il y a une marque dans la place P_2.

Les conflits sont définis de la même façon que dans le cas du modèle RdP autonome. Un exemple de conflit dans un RdP P-temporel est illustré dans la figure 2.7. On peut remarquer que dans ce cas le temps n'intervient pas dans le règlement des conflits.

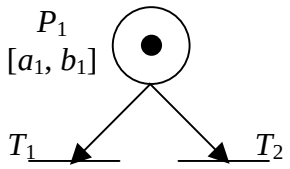


FIG. 2.7 – Conflit dans un RdP P-temporel

La puissance de modélisation de l'outil RdP P-temporel est donnée par sa capacité à représenter d'une manière simple et concise les synchronisations temporelles. Cette caractéristique de l'outil RdP P-temporel sera présentée en détail dans la sous-section 2.1.4.

Remarque 2.1. Le fonctionnement d'un RdP P-temporel considère d'abord une synchronisation temporelle (disponibilité des marquages). Ensuite on prend en compte la synchronisation logique (validation des transitions par les marques disponibles). Ceci n'est pas en concordance avec le principe de fonctionnement d'un modèle RdP, qui implique d'abord une synchronisation logique.

Remarque 2.2. Le phénomène de mort des marques dans les places, spécifique à cet outil, rend difficile l'interprétation du modèle. De manière intuitive, dans un RdP les places avec leur marquage représentent l'état du système et les transitions modélisent le changement d'état. Le passage d'une marque de l'état de disponibilité à l'état de marque morte correspond à un changement de nature différente. Cela rend plus complexe l'exploitation de ce modèle.

■

Une autre extension du modèle RdP autonome qui a retenu notre attention toujours grâce à sa capacité de modéliser les contraintes temporelles intervenant dans le fonctionnement d'un système est le modèle RdP T-temporel. Nous présentons ce modèle dans la sous-section suivante.

2.1.3 Le modèle RdP T-temporel

Le modèle RdP T-temporel a été introduit dans [MF76] et [BD91] pour la modélisation et l'analyse des systèmes de communication. Cet outil est dérivé du modèle RdP autonome en associant un intervalle du temps $[a_j, b_j]$ à chaque transition T_j. Les notions de transition validée et transition franchissable ne sont plus équivalentes dans un modèle RdP T-temporel. Une transition est validée par le marquage du RdP autonome sous-jacent. Par contre, elle peut être franchie seulement lorsqu'une durée comprise dans l'intervalle du temps associé s'est écoulée depuis l'instant de sa validation. Avant de donner la définition formelle du modèle RdP T-temporel, nous présentons son principe de fonctionnement à travers un exemple.

Exemple 2.3. Considérons à nouveau l'exemple de la machine avec stock présenté dans la section 2.1. Cette fois-ci nous introduisons des contraintes temporelles sur la date d'exécution des événements. Ainsi, l'intervalle du temps entre deux dépôts successifs d'une pièce dans le stock est de 1.1 à 1.5 u.t. De même, la durée de traitement d'une pièce par la machine est de 0.9 à 1.2 u.t. Le fonctionnement de ce système est représenté par le RdP T-temporel illustré dans la figure 2.8.

Le dépôt d'une pièce dans le stock est modélisé par le franchissement de la transition T_1. On associe l'intervalle [1.1, 1.5] à cette transition.

La transition T_2 modélise le traitement d'une pièce par la machine. On associe l'intervalle [0.9, 1.2] à cette transition. Lorsqu'une marque arrive dans la place P_2, la transition T_2 devient validée. Cependant, elle ne peut pas être franchie avant qu'une durée de 0.9 u.t. se soit écoulée depuis l'instant de sa validation. Par contre, elle doit être franchie au plus tard au bout de 1.2 u.t.

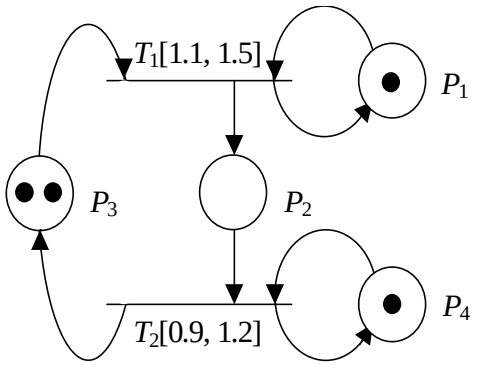


FIG. 2.8 – RdP T-temporel

■

Formellement un RdP T-temporel est défini de la manière suivante [BD91] :

Définition 2.3. Un RdP T-temporel est un 6-uplet $< P, T, Pre, Post, M_0, Is >$, tel que :

- $P = \{P_1, P_2, \ldots P_n\}$ est l'ensemble fini des places ;
- $T = \{T_1, T_2, \ldots T_m\}$ est l'ensemble fini des transitions, $P \cap T = \phi$;
- $Pré$ est l'application d'incidence avant ;
- $Post$ est l'application d'incidence arrière ;
- M_0 est le marquage initial ;

- $Is : T \rightarrow \mathbb{Q} \times (\mathbb{Q} \cup \infty)$ est une fonction qui associe un intervalle de temps à chaque transition. $\mathbb{Q}$ représente l'ensemble des nombres rationnels positifs.

$$\begin{gathered} Is(T_i) = [a_i^s,\ b_i^s] \\ a_i^s \leq b_i^s \\ 0 \leq a_i^s < \infty \\ 0 \leq b_i^s \leq \infty \end{gathered}$$

où a_i^s et b_i^s sont des nombres rationnels positifs.

■

L'intervalle $[a_i^s,\ b_i^s]$ est appelé intervalle statique de franchissement. Il modélise la contrainte temporelle imposée pour le franchissement de la transition T_i. L'origine de temps considérée pour la définition de cet intervalle est l'instant de validation de la transition T_i.

- a_i^s est l'instant de franchissement au plus tôt ;
- b_i^s est l'instant de franchissement au plus tard.

L'état d'un RdP T-temporel est défini par le couple S=(M, I) [BD91], où :

- M représente le marquage du RdP autonome sous-jacent ;
- I est le vecteur des intervalles de franchissement. Chacun de ces éléments mémorise les dates possibles pour le franchissement d'une transition validée par le marquage.

Une transition peut être franchie si les conditions suivantes sont vérifiées :

- ***Condition de marquage*** : La transition doit être validée par le marquage du RdP autonome sous-jacent ;
- ***Condition de temps*** : Une transition validée ne peut pas être franchie avant l'écoulement d'une période égale à son instant de franchissement au plus tôt depuis l'instant de sa validation. Par contre, elle doit être franchie avant une période de temps égale à son instant de franchissement au plus tard.

Les notions de transition validée et transition franchissable ne sont pas équivalentes dans le modèle RdP T-temporel. Une transition est validée si elle vérifie la condition de marquage. Elle est franchissable lorsque les deux conditions sont vérifiées.

Remarque 2.3. Le fonctionnement d'un modèle RdP T-temporel considère d'abord une synchronisation logique (validation des transitions par le marquage). Une fois cette synchronisation est réalisé (validation de certaines transitions) on prend en compte les contraintes temporelles. Ceci respecte le principe de fonctionnement de l'outil RdP.

Remarque 2.4. Contrairement à l'outil RdP P-temporel, l'évolution d'un RdP T-temporel est déterminée par l'occurrence d'un seul type d'événements : le franchissement des transitions.

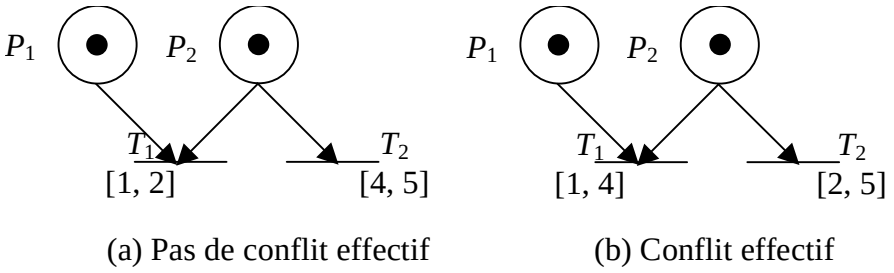
■

Un conflit apparaît lorsque le marquage d'une place ne permet pas de franchir toutes ses transitions de sortie validées. Contrairement au modèle RdP P-temporel, la règle de fonctionnement d'un RdP T-temporel fait que les intervalles de franchissement associés aux transitions du conflit peuvent résoudre le conflit.

Considérons le RdP T-temporel présenté dans la figure 2.9.a. Les transitions T_1 et T_2 sont validées en même temps par le marquage des places P_1 et P_2. Considérons que chaque place contient une marque et que la marque dans P_2 est arrivée après la marque dans P_1. Dans ce cas, les transitions T_1 et T_2 sont validées en même temps. Le nombre de marques dans ces places ne suffit pas pour franchir les deux transitions. Par conséquent,

ces transitions sont en conflit. Cependant, selon les intervalles de franchissement associés aux transitions, T_1 doit être franchie au plus tard à 2 u.t., tandis que T_2 ne peut pas être franchie avant 4 u.t. Ainsi, la transition T_1 sera toujours franchie au détriment de la transition T_2. Dans ce cas, le conflit est résolu par les intervalles de franchissement associés aux transitions T_1 et T_2.

Considérons maintenant le RdP T-temporel présenté dans la figure 2.9.b. Dans ce cas, selon les intervalles de franchissement associés, les transitions T_1 et T_2 sont toutes les deux franchissables entre 2 et 4 u.t. depuis l'instant de validation. Par conséquent, ce conflit n'est plus résolu par les intervalles de franchissement associés aux transitions. Un exemple de politique de résolution pour ce conflit est de donner la priorité au franchissement de la transition T_1 par rapport à la transition T_2.



(a) Pas de conflit effectif (b) Conflit effectif

FIG. 2.9 – Conflits dans un RdP T-temporel

Le modèle RdP T-temporel permet de représenter des comportements où un choix est forcé ou plusieurs choix sont possibles.

Les outils RdP P-temporel et RdP T-temporel sont tous les deux bien appropriés pour la modélisation des SEDT dont le fonctionnement est sujet à des contraintes temporelles fortes (ex : les tâches ont une durée minimale et une durée maximale). Cependant, chacun de ces outils a ses spécificités qui le font plus au moins adapté selon l'application envisagée.

Dans notre travail, nous nous intéressons à la synthèse de la commande par supervision pour les SEDT. L'outil qui nous est apparu le plus approprié pour la modélisation des SEDT à commander est le modèle RdP T-temporel. Pour justifier notre choix, nous faisons une brève comparaison des outils RdP P-temporel et RdP T-temporel dans la sous-section suivante.

2.1.4 Comparaison des outils RdP T-temporel et RdP P-temporel

Il y a deux différences entre les modèles fournis par les outils RdP P-temporels et les RdP T-temporels : le traitement des synchronisations temporelles et l'interprétation des conflits.

Considérons d'abord la modélisation des synchronisations temporelles par ces deux outils.

Le modèle fourni par l'outil RdP P-temporel pour représenter une synchronisation temporelle est illustré dans la figure 2.10. La validation de la transition T_1 est conditionnée par la présence d'au moins une marque disponible dans chacune de ses places d'entrée P_1 et P_2. L'intervalle de temps associé à la place P_1 est $[a_1, b_1]$. Selon la règle de fonctionnement d'un RdP P-temporel, une marque déposée dans P_1 est disponible pour la validation de T_1 lorsqu'elle y a séjourné au moins a_1 u.t. Par contre, la marque devient morte au delà de b_1.

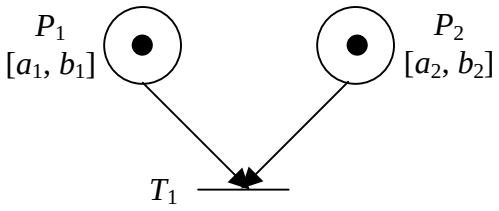


FIG. 2.10 – Synchronisation temporelle modélisée par un RdP P-temporel

Le même comportement peut être modélisé par le RdP T-temporel illustré dans la figure 2.11.

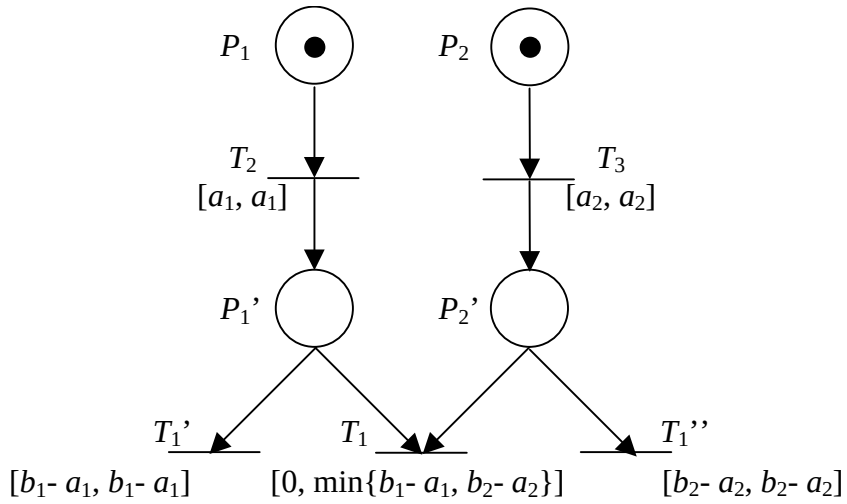


FIG. 2.11 – Modèle RdP T-temporel équivalent

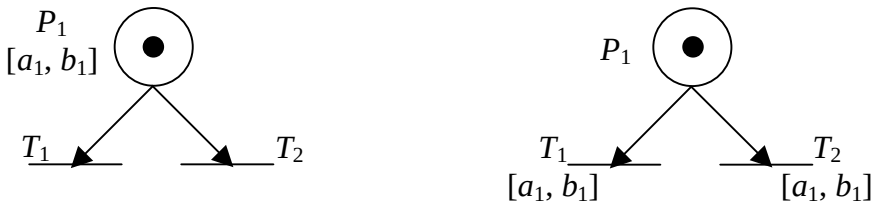
Une marque dans P_1 devient disponible pour la validation de T_1 après y avoir séjourné a_1 u.t. Cet événement est modélisé par le franchissement de la transition T_2. L'intervalle de franchissement associé à cette transition est $[a_1, a_1]$. Suite au franchissement de T_2, la marque est retirée de P_1 et déposée dans P_1'. Cette marque reste disponible pour la validation de la transition T_1 pendant une durée égale à $b_1 - a_1$ u.t. Une fois cette durée dépassée, la transition T_1' est franchie et la marque est enlevée de la place P_1'. Ainsi elle ne peut plus jamais participer à la validation de la transition T_1. Par conséquent, le franchissement de la transition T_1' dans le modèle RdP T-temporel est équivalent à la mort d'une marque dans le modèle RdP P-temporel.

De même, lorsque les transitions T_1', T_1 et T_1" sont en conflit effectif, on donne la priorité au franchissement de la transition T_1.

Considérons maintenant l'interprétation des conflits dans les modèles fournis par les outils RdP P-temporel et RdP T-temporel.

Le modèle fournit par l'outil RdP P-temporel pour représenter un conflit est illustré dans la figure 2.12.a. Considérons qu'il y a une seule marque dans la place P_1. Une fois que cette marque y a séjourné pendant a_1 u.t., les transitions T_1 et T_2 deviennent validées. Par contre, elles ne peuvent pas être franchies toutes les deux. On a un conflit effectif.

Le même comportement peut être modélisé par le RdP T-temporel illustré dans la figure 2.12.b. Ce modèle est obtenu en associant aux transitions T_1 et T_2 l'intervalle de temps correspondant à la place P_1 dans le modèle RdP P-temporel.



(a) Conflit dans un RdP P-temporel

(b) Modèle équivalent RdP T-temporel

FIG. 2.12 – Modèles RdP P-temporel et RdP T-temporels équivalents

Lorsqu'il y a un conflit dans un modèle RdP T-temporel, le temps peut déterminer le choix de la transition à franchir. Ce n'est pas le cas dans un modèle RdP P-temporel.

Considérons le conflit modélisé par le RdP T-temporel illustré dans la figure 2.13.

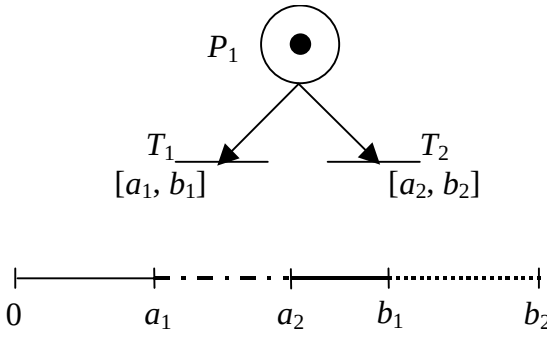


FIG. 2.13 – Conflit dans un RdP T-temporel

Lorsqu'une marque arrive en P_1, on commence à compter le temps pour les transitions T_1 et T_2. Lorsque la durée écoulée est comprise entre a_1 et a_2 u.t., seulement la transition T_1 est franchissable. Par contre, après a_2 u.t. la transition T_2 devient elle aussi franchissable. Cependant, T_1 doit être franchie au plus tard lorsque la durée pendant laquelle elle est restée validée atteint la valeur b_1. Ce comportement ne peut pas être représenté d'une manière simple par un modèle RdP P-temporel.

En conclusion, l'outil RdP P-temporel est plus approprié pour la représentation des synchronisations temporelles. Par contre, l'outil RdP T-temporel offre plus de possibilités pour la modélisation des conflits.

Selon la remarque 2.1, le modèle RdP P-temporel met l'accent sur la synchronisation temporelle. Ceci ne correspond pas au principe de fonctionnement des modèles RdP, qui considère d'abord une synchronisation logique. Par contre, selon la remarque 2.3, le fonctionnement d'un modèle RdP T-temporel respecte la philosophie des outils RdP.

De plus, l'interprétation d'un modèle RdP T-temporel est plus intuitive par rapport à un modèle RdP P-temporel parce que son évolution est déterminée par l'occurrence d'un seul type d'événements, i.e. le franchissement des transitions.

Ces arguments font que nous avons retenu l'outil RdP T-temporel pour la modélisation des SEDT en vue de la synthèse de la commande par supervision.

Par la suite nous présentons la compositions synchrone des RdP T-temporels. Cette opération nous est utile pour obtenir le modèle du procédé avec spécification à partir des modèles RdP T-temporel du procédé sans spécification et de la spécification imposée sur son fonctionnement.

2.1.5 Composition synchrone des RdP T-temporels

La définition de cette opération est inspirée de [Kou99] où l'auteur décrit la composition synchrone des RdP à arcs temporels.

Le principe de la composition synchrone des RdP T-temporels est basé sur le fait que chaque transition modélise l'occurrence d'un événement. La condition de validation d'une transition d'un RdP T-temporel modélise les contraintes logiques qui doivent être satisfaites pour que l'occurrence de l'événement associé soit possible. De plus, la contrainte temporelle sur la date d'occurrence de l'événement est modélisée par l'intervalle de franchissement associé à la transition correspondante.

La composition synchrone des RdP T-temporels consiste dans la fusion des transitions identiques (associées au même événement). L'intervalle de franchissement associé à la transition obtenue par la fusion est défini par l'intersection des intervalles de franchissement associés aux transitions fusionnées. Cette opération de synchronisation est structurelle. Elle est indépendante du marquage. Ceci constitue un grand avantage des RdP. Dans un automate, cette opération est dynamique et donc plus complexe.

Nous présentons le principe de la composition synchrone des RdP T-temporels à travers un exemple.

Exemple 2.4. Considérons le poste de collage présenté dans la section 1.2. Cette fois ci, nous considérons que la date d'exécution de chaque événement est soumise à des contraintes temporelles.

On suppose que le robot doit commencer sa tâche au plus tard à 2 u.t. depuis la date de démarrage du processus de collage. De plus, il a besoin de 4 à 5 u.t. pour transporter la pièce O_2. Ensuite il la dépose tout de suite sur la pièce O_1 en vue de réaliser le collage. Si la colle est prête pour le collage au moment du dépôt de la pièce O_2, alors le collage est réalisé. Sinon, il y a un échec.

En ce qui concerne la tâche de l'opérateur, on suppose qu'il n'y a aucune contrainte sur sa date de démarrage. Par contre, l'opérateur a besoin de 1 u.t. pour l'accomplir. De plus, pour obtenir un collage de qualité il faut que la colle ne soit ni trop sèche, ni trop humide. On suppose que la colle est prête pour le collage après 3 u.t. depuis le début de la tâche de l'opérateur. Elle reste propre pour le collage pendant 3.5 u.t. Après elle devient trop sèche.

La spécification imposée au procédé exige que l'opérateur finisse sa tâche avant que le robot dépose la pièce O_2 pour le collage.

Par la suite, nous présentons les modèles RdP T-temporels du procédé, de la spécification et du comportement désiré du procédé.

Le déroulement de la tâche du robot est modélisé par le RdP T-temporel illustré dans la figure 2.14.

Lorsqu'une marque est présente dans la place P_1, le robot est prêt pour commencer sa tâche. Cet événement est modélisé par le franchissement de la transition T_1. Selon la contrainte temporelle imposée sur la date de lancement de sa tâche, le robot doit commencer cette tâche au plus tard 2 u.t. après le démarrage du processus de collage. Par conséquent, l'intervalle de franchissement associé à T_1 est [0, 2]. La présence d'une marque dans la place P_2 modélise le déroulement du transport de la pièce O_2. La durée de cette opération est de 4 à 5 u.t. Son accomplissement est modélisé par le franchissement de la transition T_2. La présence d'une marque dans P_3 modélise le fait que le transport de la pièce O_2 est achevé et elle peut être déposée. Il y a deux évolutions possibles. Soit il y a un collage, soit un échec. L'échec est modélisé par le franchissement de la transition T_3, tandis que le collage par le franchissement de T_4'. On a supposé que le robot dépose la

pièce tout de suite. Ainsi, l'intervalle de franchissement associé à chacune des transitions T_3 et T_4' est [0, 0]. Le franchissement de la transition T_3 est une évolution non-désirée.

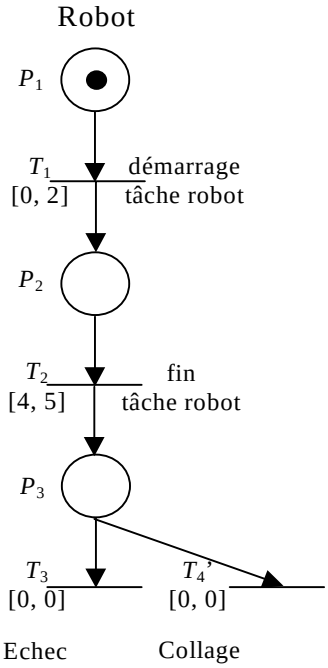


FIG. 2.14 – Modèle RdP T-temporel de la tâche du robot

Le RdP T-temporel qui modélise la tâche de l'opérateur est illustré dans la figure 2.15.

Lorsqu'une marque est dans P_4, l'opérateur peut commencer sa tâche, i.e. la préparation de la pièce O_1 pour le collage. Cet événement est modélisé par le franchissement de la transition T_5. Il n'y a aucune contrainte temporelle sur l'instant de démarrage de cette tâche. Ainsi, l'intervalle de franchissement associé à T_5 est $[0, \infty)$. La préparation de la pièce O_1 est modélisée par la présence d'une marque dans la place P_6. Cette opération prend 1 u.t. et son accomplissement est modélisé par le franchissement de la transition T_6. L'intervalle de franchissement associé à cette transition est [1, 1].

La colle devient bonne pour le collage après 3 u.t. depuis le démarrage de la tâche de l'opérateur. Lorsqu'une marque est dans P_5, la colle a été déposée par l'opérateur. Il faut attendre 3 u.t. pour qu'elle devienne prête pour le collage. Cet événement est modélisé par le franchissement de T_7. L'intervalle de franchissement associé à T_7 est [3, 3]. La présence d'une marque dans P_7 modélise le fait que la colle est prête pour le collage. Il y a deux possibilités d'évolution. Soit il y a un collage, soit la colle devient trop sèche, donc il y a un échec.

La réalisation d'un collage est modélisée par le franchissement de la transition T_4".

On sait que la colle reste propre pour le collage pendant 3.5 u.t. Ainsi, l'intervalle de franchissement associé à T_4" est [0, 3.5]. Le fait que la colle devient trop sèche est modélisé par le franchissement de T_8. L'intervalle de franchissement associé à cette transition est [3.5, 3.5]. Le franchissement de la transition T_8 est une évolution non-désirée.

Le modèle du comportement du procédé sans spécification est obtenu par la composition synchrone des RdP T-temporels modélisant la tâche du robot et respectivement la tâche de l'opérateur. Ce modèle est présenté dans la figure 2.16. Il a été obtenu par la fusion des transitions T_4' et T_4" qui modélisent l'occurrence de l'événement *collage* dans les deux sous-modèles.

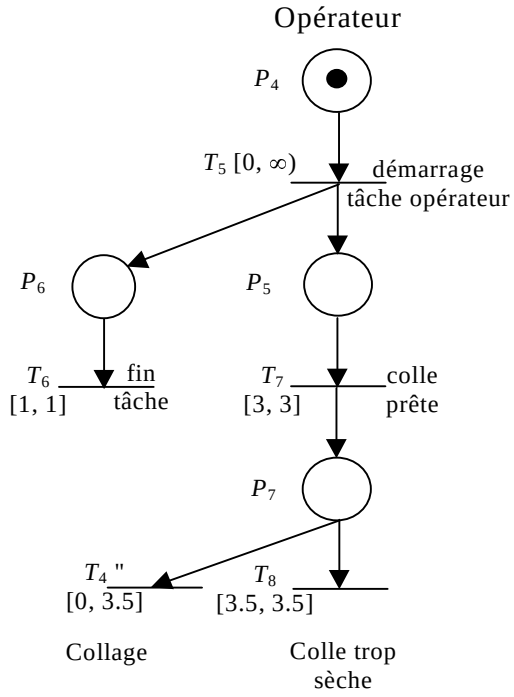


FIG. 2.15 – Modèle RdP T-temporel de la tâche de l'opérateur

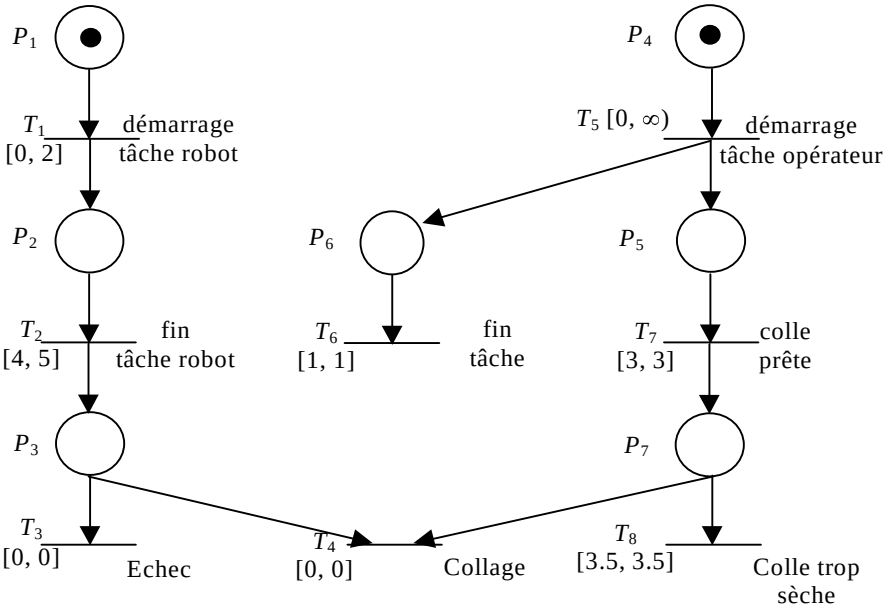


FIG. 2.16 – Modèle RdP T-temporel du procédé à commander

Soit T_4 la transition obtenue par la fusion de T_4' et T_4". Par la suite il faut déterminer l'intervalle de franchissement associé à cette transition.

Dans le procédé, l'événement *collage* peut arriver dès que le robot finit sa tâche et une durée comprise entre 0 et 3.5 u.t. s'est écoulée depuis l'instant où la colle est devenue prête pour le collage. Ainsi, l'intervalle de franchissement associé à la transition T_4 est [0, 0], i.e. la conjonction des intervalles [0, 0] et [0, 3.5] associés à T_4' et respectivement T_4".

La spécification est modélisée par le RdP T-temporel illustré dans la figure 2.17.

Le franchissement de la transition T_6 modélise l'achèvement de la tâche de l'opérateur,

tandis que le franchissement de T_4 représente le réalisation du collage. Initialement la place P_8 ne contient aucune marque. Ainsi, la transition T_4 est toujours franchie après T_6. Il n'y a aucune contrainte temporelle sur la succession des événements *fin tâche opérateur* et *fin tâche robot*. Ainsi, les intervalles de temps associés aux transitions T_6 et T_4 ont la valeur $[0, \infty)$.

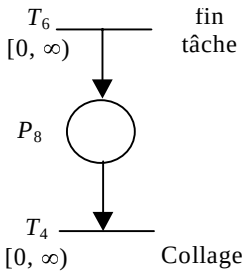


FIG. 2.17 – Modèle RdP T-temporel de la spécification

Le comportement désiré du procédé est obtenu par la composition synchrone des RdP T-temporels modélisant le comportement du procédé sans spécification et, respectivement, la spécification. Le modèle RdP T-temporel du comportement désiré du processus de collage est présenté dans la figure 2.18. Lorsque notre objectif est d'obtenir un collage réussi, le franchissement de la transition T_4 est prioritaire par rapport à aux transitions T_3 et T_8.

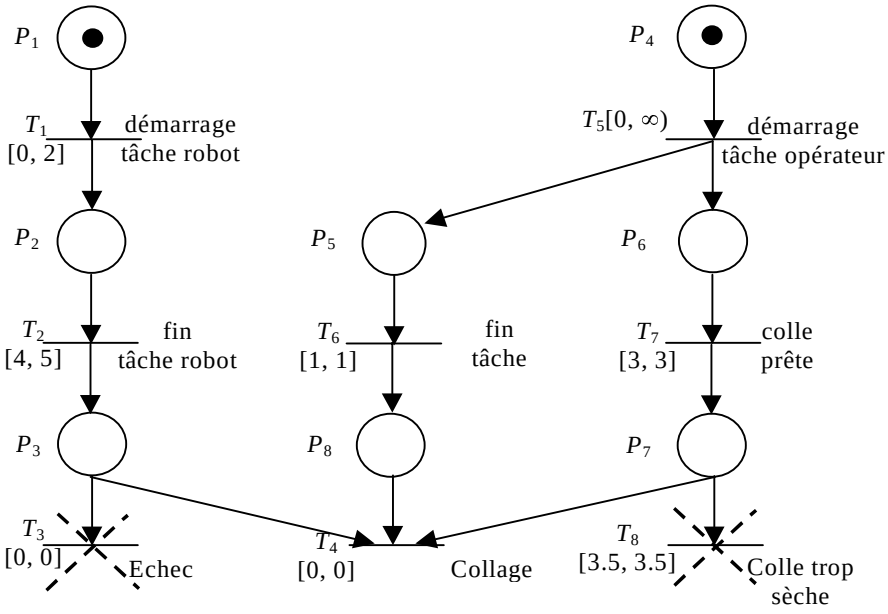


FIG. 2.18 – Modèle RdP T-temporel du comportement désiré du procédé

■

Après avoir modélisé le comportement du procédé avec spécification par un RdP T-temporel, il faut analyser le comportement de ce réseau afin de détecter l'existence d'évolutions non désirés. L'approche proposée dans la littérature pour l'analyse des propriétés d'un RdP T-temporel est basée sur la construction du graphe des classes d'états

[BD91] [Gal97] [JAGZ00]. Nous considérons que l'outil automate temporisé est plus approprié pour la modélisation et l'analyse d'un RdP T-temporel en vue de la synthèse de la commande. Nous avons présenté une comparaison de ces deux outils dans [Sav00]. Généralement, la taille du modèle fourni par l'outil automate temporisé est moins importante. De plus, cet outil offre des techniques puissantes d'analyse d'atteignabilité des états.

Nous présentons le modèle automate temporisé dans la section suivante.

2.2 L'outil automate temporisé

Dans cette section nous présentons l'outil automate temporisé. Une attention particulière sera portée au calcul des successeurs et des prédécesseurs d'une région d'horloges [Yov98]. Ces procédures sont à la base de notre approche de la commande.

2.2.1 Présentation du modèle automate temporisé

Un modèle automate temporisé est défini comme étant une machine à états finis munie d'un ensemble de variables continues par morceaux, appelées horloges [AD94] [Yov93]. Ces variables servent à compter le temps écoulé. Elles sont incrémentées uniformément avec le passage du temps lorsque le système séjourne dans un sommet. Les horloges sont synchronisées et avancent avec le même pas. La dynamique d'une horloge x dans un sommet de l'automate temporisé est décrite par l'équation $\dot{x} = 1$. A chaque sommet de l'automate on associe un prédicat sur la valeur des horloges appelé *invariant du sommet*. L'automate peut séjourner dans un sommet tant que l'invariant correspondant est vérifié par la valeur des horloges.

Le franchissement d'une transition est instantané. Il peut déterminer la mise à zéro des certaines horloges. Ce changement discret de la valeur d'une horloge x_i est modélisé par la relation $x_i := 0$. L'ensemble des horloges qui sont mises à zéro lors du franchissement d'une transition est décrit par une *affectation*. De plus, à chaque transition on associe un prédicat sur la valeur des horloges, appelé *garde*. Ce prédicat détermine les dates possibles pour le franchissement de la transition. Ainsi, une transition de l'automate temporisé peut être franchie seulement si sa garde est vérifiée par la valeur des horloges. Les gardes modélisent les contraintes temporelles imposées sur le fonctionnement du système.

Nous illustrons les concepts de base, ainsi que la notation graphique du modèle automate temporisé à travers un exemple de modélisation d'un distributeur de boissons [Oli94].

Exemple 2.5. Le distributeur de boissons que nous considérons permet de choisir du thé ou du café. Lorsque l'utilisateur introduit une pièce, il dispose de dix secondes pour faire son choix. Si au bout de cette période il n'a rien choisi, le distributeur lui rend la pièce et redevient disponible. L'automate temporisé qui modélise le comportement de ce système est représenté dans la figure 2.19.

Cet automate temporisé est composé d'un ensemble de sommets $\{L_0, L_1\}$ reliés par des transitions. Il est équipé d'une seule horloge, x, utilisée pour mesurer le temps écoulé depuis l'introduction d'une pièce. Lorsque le système séjourne dans un sommet de l'automate, la dynamique de cette horloge est décrite par l'équation $\dot{x} = 1$.

Le sommet initial de l'automate est L_0. Il est marqué par une flèche entrante sur laquelle on marque la valeur initiale des horloges. Le sommet L_0 modélise l'état de disponibilité du distributeur. Lorsque le système séjourne dans L_0, le distributeur attend

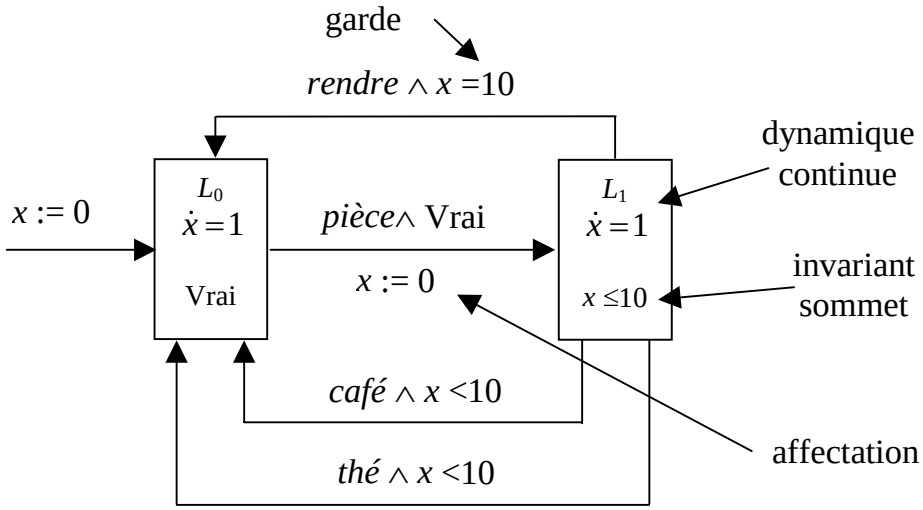


FIG. 2.19 – Automate temporisé qui modélise le distributeur de boissons

que l'utilisateur introduise une pièce. Il n'y a aucune contrainte temporelle sur la durée d'attente d'une pièce. Ainsi, le système peut séjourner dans ce sommet pendant un temps indéfini, donc l'invariant du sommet associé à L_0 a la valeur "Vrai".

Lorsqu'un utilisateur introduit une pièce, l'événement *pièce* se produit et le système franchit la transition allant du L_0 vers L_1. Il n'y a aucune contrainte temporelle sur l'instant de dépôt d'une pièce par l'utilisateur. Par conséquent, la garde associée à cette transition a la valeur "Vrai". L'affectation $x := 0$ associée à cette transition modélise l'initialisation de l'horloge x par son franchissement.

Lorsque le système se trouve dans le sommet L_1, le distributeur attend que l'utilisateur fasse son choix. La valeur de l'horloge x dans le sommet L_1 est égale au temps écoulé depuis le dépôt de la pièce par l'utilisateur. L'invariant du sommet associé à L_1 est $x \leq 10$. Ainsi, le système doit quitter le sommet L_1 au plus tard 10 secondes après y être arrivé. Lorsque $x < 10$ seulement les transitions associées aux événements *thé* et *café* peuvent être franchies parce que toutes les deux ont la garde $x < 10$. Lorsque x atteint la valeur 10, on franchit la transition associée à l'événement *rendre*, parce qu'elle a la garde $x = 10$. Le cycle se répète lorsque le système retourne au sommet L_0.

■

Avant de donner la définition formelle de l'outil automate temporisé, nous présentons quelques définitions concernant les horloges [Alu99].

Soit X un ensemble fini des horloges.

Définition 2.4. Soit x_1, $x_2 \in X$ des horloges, $c \in \mathbb{Q}$ une constante rationnelle et $\prec \in \{<, \leq, \geq\}$ une relation d'ordre. Une contrainte g sur la valeur des horloges est définie par une des expressions suivantes :

- $g := x_1 \prec c$,
- $g := x_1 - x_2 \prec c$,
- $g := g_1 \wedge g_2$,
- $g := \neg g_1$.

où g_1 et g_2 sont des contraintes sur la valeur des horloges.

■

L'ensemble des contraintes sur la valeur des horloges est noté G_X.

Définition 2.5. Une valuation des horloges est une fonction $v : X \to \mathbb{R}^+$, qui affecte un nombre réel positif à chaque horloge.

■

Ainsi, une valuation des horloges définit la valeur de toutes les horloges à un instant donné. L'ensemble des valuations des horloges $x \in X$ est noté V_X. La notation $v \models g$ exprime le fait que la valuation $v \in V_X$ vérifie la contrainte $g \in G_X$.

Définition 2.6. Une affectation, notée $A_{m,n}$, désigne l'ensemble des horloges qui seront mises à zéro par le franchissement d'une transition $T_{m,n}$. Elle est décrite par un ensemble d'équations de type $x_i := 0$.

■

Soit $v \in V_X$ une valuation, $t \in \mathbb{R}^+$ un nombre réel et $A_{m,n}$ une affectation. On note avec $v + t$ la valuation qui affecte la valeur $v(x) + t$ à chaque horloge $x \in X$.

De même, on note avec $v[A_{m,n}]$ la valuation qui met à zéro les horloges spécifiées par l'affectation $A_{m,n}$ et laisse inchangée la valeur des autres horloges.

Formellement, un automate temporisé est défini de la façon suivante [Alu99] [AD94] [Yov93] :

Définition 2.7. Un automate temporisé est un 6-uplet $\mathcal{A}$=($\mathcal{L}$, L_0, X, $\sum$, I, $\mathcal{T}$), où :

- $\mathcal{L}$ est l'ensemble fini des sommets ;
- $L_0 \subset \mathcal{L}$ est le sommet initial ;
- X est l'ensemble fini des horloges ;
- $\sum$ est un ensemble de symboles ;
- I est une application qui associe un invariant du sommet $I(L_m)$ à chaque sommet $L_m \subset \mathcal{L}$;
- $\mathcal{T}$ est l'ensemble des transitions. Une transition est un 5-uplet (L_m, a, $g_{m,m+1}$, $A_{m,m+1}$, L_{m+1}), où :
  - L_m est le sommet source ;
  - $a \in \sum$ est un symbole associé à un événement ;
  - $g_{m,m+1}$ est la condition de franchissement, i.e. la garde ;
  - $A_{m,m+1}$ est l'affectation ;
  - L_{m+1} est le sommet destination.

■

Définition 2.8. L'état d'un automate temporisé est défini par le couple (L_m, v), où L_m désigne le sommet et v est une valuation d'horloges qui vérifie l'invariant du sommet $I(L_m)$.

■

On peut séjourner dans un sommet de l'automate temporisé tant que l'invariant associée est satisfait par la valeur des horloges. Alors un sommet peut ne pas correspondre à un seul état mais à un espace d'état. Les valeurs que les horloges peuvent prendre pendant le séjour du système dans un sommet décrivent un espace d'horloges. On appelle région dans un sommet et on la note (L_m, Q_n) un espace d'horloges Q_n dans un sommet L_m.

Généralement, l'étude d'un système modélisé par un automate temporisé est basée sur l'analyse d'atteignabilité des états de l'automate. Pour savoir si une région Q est atteignable depuis une région Q_{init}, deux méthodes peuvent être utilisées.

La première méthode est basée sur le calcul de l'espace de tous les états qui peuvent être atteints depuis des états appartenant à la région Q_{init}. L'ensemble de ces états est

appelé successeur de la région Q_{init}. Si l'espace calculé contient des états qui appartiennent également à la région Q, alors on peut conclure que cette région est atteignable depuis Q_{init}. Cette méthode est appelée *méthode d'analyse en avant.*

La deuxième méthode est basée sur le calcul de l'ensemble de tous les états à partir desquels on peut atteindre des états de la région Q. L'ensemble de ces états est appelé prédécesseur de la région Q. Si l'espace calculé contient des états qui appartiennent également à la région Q_{init}, alors on peut conclure que la région Q est atteignable depuis Q_{init}. Cette méthode, duale à la méthode d'analyse en avant, est appelée *méthode d'analyse en arrière.* Les deux méthodes sont détaillées dans [Alu99] [Yov93] [ACH+95]. Ces procédures d'analyse d'atteignabilité ont été implémentés dans des logiciels dédiés à la vérification des systèmes temporisés et hybrides. Parmi ces logiciels nous citons Kronos [Yov97] [COTY96] et Hytech [HHWT95] [Ho95] [Hen96].

Dans notre travail, nous nous intéressons seulement aux procédures de calcul des successeurs et des prédécesseurs d'une région [Yov98], que nous présentons par la suite.

2.2.2 Calcul des successeurs d'une région

Un automate temporisé peut avoir deux possibilités d'évolution à partir d'un état. Il peut rester dans le même sommet en laissant le temps s'écouler, ou il peut y avoir franchissement d'une transition de sortie. Par conséquent, un état d'un automate temporisé peut avoir deux types de successeurs : continus et discrets.

Un *successeur continu* d'un état est obtenu en restant dans le même sommet et en laissant le temps évoluer.

Définition 2.9. L'état $(L_m,\ v+t)$ est un successeur continu de l'état $(L_m,\ v)$ si :

$$\exists t \in \mathbb{R}^+ \text{ t.q.} \forall t' \leq t,\ v+t' \models I(L_m)$$
$$(L_m, v) \rightarrow (L_m, v+t).$$

■

Ceci veut dire que les horloges sont incrémentées pendant une durée t, tout en restant dans le sommet L_m. L'invariant $I(L_m)$ du sommet L_m est vérifié par la valeur des horloges tout au long de cette période $(\forall t' \leq t,\ v+t' \models I(L_m))$.

Un *successeur discret* d'un état est obtenu en franchissant une transition de l'automate.

Définition 2.10. L'état $(L_{m+1},\ v')$ est le successeur discret de l'état $(L_m,\ v)$ par le franchissement de la transition $T_{m,m+1}=(L_m,\ a,\ g_{m,m+1},\ A_{m,m+1},\ L_{m+1})$ si :

$$(L_m,\ a,\ g_{m,m+1},\ A_{m,m+1},\ L_{m+1}) \in \mathcal{T} \wedge (v \models g_{m,m+1}) \wedge$$
$$\wedge (v' = v[A_{m,m+1}]) \wedge (v[A_{m,m+1}] \models I(L_{m+1}))$$
$$(L_m, v) \rightarrow (L_{m+1}, v')$$

■

En effet, une transition peut être franchie seulement si sa garde est vérifiée par la valeur des horloges $(v \models g_{m,m+1})$. Lors du franchissement d'une transition la valeur des horloges est modifiée selon l'affectation associée $(v' = v[A_{m,m+1}])$. De plus, la valeur des horloges après le franchissement doit vérifier l'invariant du sommet destination $(v[A_{m,m+1}] \models I(L_{m+1}))$.

De la même manière que pour les états, on peut définir les successeurs d'une région.

Définition 2.11. L'ensemble des états atteignables à partir de tout état $(L_m,\ v) \in (L_m,\ Q_n)$ en laissant le temps s'écouler tout en restant dans le même sommet est appelé successeur continu de la région $(L_m,\ Q_n)$. Cet ensemble, noté $Suc_t(Q_n)$, est défini par l'expression :

$$v' \models Suc_t(Q_n) \text{ ssi } \exists t \in \mathbb{R},\ v' - t \models Q_n \wedge \forall t' \in \mathbb{R}^+,\ t' \leq t \Rightarrow v' - t' \models I(L_m).$$

■

Exemple 2.6. Soit l'automate temporisé illustré dans la figure 2.20.

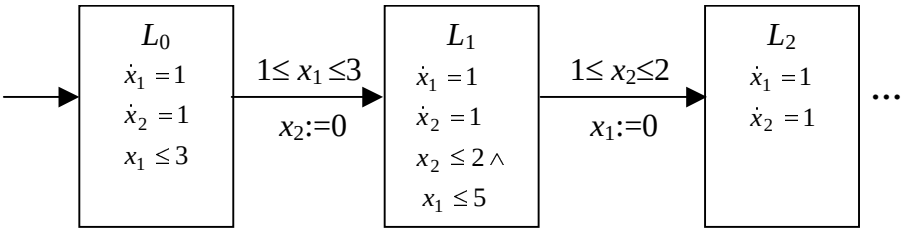


FIG. 2.20 – Automate temporisé

Soit (L_1, Q_1) une région dans le sommet L_1. L'espace Q_1, représenté dans la figure 2.21.a, est décrit par les contraintes suivantes sur la valeur des horloges :

$$Q_1 = [1 \leq x_1 \leq 3 \wedge 0 \leq x_2 \leq 1]$$

Avant de calculer le successeur continu de cette région, nous rappelons que l'évolution d'une horloge x_i est décrite par l'équation $\dot{x}_i = 1$. De même, le système peut séjourner dans le sommet L_1 tant que l'invariant $I(L_1)$ est satisfait par la valeur des horloges, i.e. $0 \leq x_2 \leq 2$. Le successeur continu de la région (L_1, Q_1) est :

$$\begin{aligned} Suc_t(Q_1) &= \exists t \in \mathbb{R}^+ . Q_1[x_1 - t, x_2 - t].I(L_1) \\ &= \exists t \in \mathbb{R}^+ . [0 \leq t \wedge 1 \leq x_1 - t \leq 3 \wedge 0 \leq x_2 - t \leq 1 \wedge 0 \leq x_2 \leq 2] \\ &= [0 \leq x_1 - x_2 \leq 3 \wedge 1 \leq x_1 \wedge 0 \leq x_2 \leq 2] \end{aligned}$$

Les variables x_1 et x_2 désignent la valeur des horloges dans l'espace $Suc_t(Q_1)$. Selon la définition d'un successeur continu, $(x_1, x_2) \in Suc_t(Q_1)$ si $\exists t \in \mathbb{R}^+$ tel que $(x_1 - t, x_2 - t) \in Q_1$. Ainsi, le calcul de l'espace $Suc_t(Q_1)$ est effectué de la manière suivante :

- d'abord on remplace l'occurrence de chaque variable x_i dans les inégalités qui décrivent l'espace d'horloges Q_1 par $x_i - t$;
- ensuite on élimine la variable t.

L'espace $Suc_t(Q_1)$ est illustré dans la figure 2.21.b.

■

Définition 2.12. Soit (L_m, Q_n) une région et $T_{m,m+1} = (L_m, a, g_{m,m+1}, A_{m,m+1}, L_{m+1})$ une transition. L'ensemble des états atteignables depuis tout état $(L_m, v) \in (L_m, Q_n)$ en franchissant la transition $T_{m,m+1}$ est appelé successeur discret de la région (L_m, Q_n). Cet ensemble, noté $Suc_{m,m+1}(Q_n)$ est défini par l'expression :

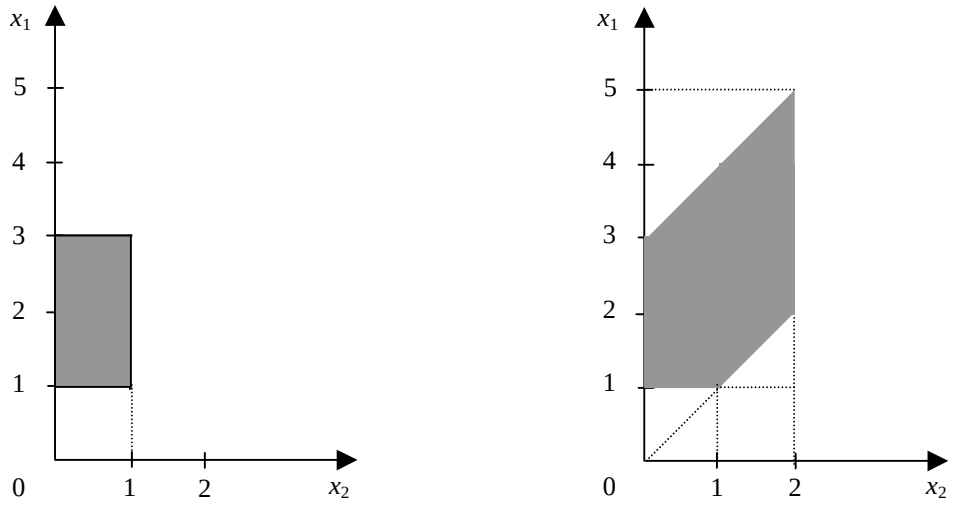
$$v' \models Suc_{m,m+1}(Q_n) \text{ ssi } \exists v \in V_X . v \models (Q_n \wedge g_{m,m+1}) \wedge v' = v[A_{m,m+1}].$$

■

Exemple 2.7. Soit la région (L_1, Q_2) dans le sommet L_1 de l'automate temporisé présenté dans la figure 2.20. L'espace Q_2, représenté dans la figure 2.22.a, est :

$$Q_2 = [0 \leq x_1 - x_2 \leq 3 \wedge 1 \leq x_1 \wedge 1 \leq x_2 \leq 2].$$

Par la suite, nous calculons l'espace $Suc_{1,2}(Q_2)$ qui est le successeur discret de cette région par le franchissement de la transition $T_{1,2} = (L_1, , 1 \leq x_2 \leq 2, x_1 := 0, L_2)$. Cette transition peut être franchie seulement si sa garde, $g_{1,2} = [1 \leq x_2 \leq 2]$, est vérifiée par



a. Espace Q_1 b. Espace $Suc_t\,(Q_1)$

FIG. 2.21 – Successeur continu d'une région

la valeur des horloges. Lors du franchissement de cette transition, la valeur de l'horloge x_2 reste inchangée, tandis que la valeur de l'horloge x_1 est modifiée par l'affectation $A_{1,2}$=$[x_1 := 0]$ associée à cette transition. Le successeur discret de la région (L_1, Q_2) est :

$$\begin{aligned}
Suc_{1,2}(Q_2) &= \exists x'_1, x'_2 . Q_2[x'_1, x'_2] \wedge x_1 = 0 \wedge x_2 = x'_2 \\
&= \exists x'_1, x'_2 . [0 \leq x'_1 - x'_2 \leq 3 \wedge 1 \leq x'_1 \wedge 1 \leq x'_2 \leq 2 \wedge x_1 = 0 \wedge x_2 = x'_2] \\
&= \exists x'_1 . [0 \leq x'_1 - x_2 \leq 3 \wedge 1 \leq x'_1 \wedge 1 \leq x_2 \leq 2 \wedge x_1 = 0] \\
&= [-3 \leq x_2 - x'_1 \leq 0 \wedge 1 \leq x'_1 \wedge 1 \leq x_2 \leq 2 \wedge x_1 = 0] \\
&= [-3 + x'_1 \leq x_2 \leq 0 + x'_1 \wedge 1 \leq x'_1 \wedge 1 \leq x_2 \leq 2 \wedge x_1 = 0] \\
&= [1 \leq x_2 \leq 2 \wedge x_1 = 0]
\end{aligned}$$

Les variables x_1 et x_2 représentent la valeur des horloges dans l'espace $Suc_{1,2}(Q_2)$, i.e. après le franchissement de la transition $T_{1,2}$. Par contre, les variables x'_1 et x'_2 désignent la valeur ancienne des horloges, i.e. avant le franchissement de $T_{1,2}$. L'espace $Suc_{1,2}(Q_2)$ est illustré dans la figure 2.22.b.

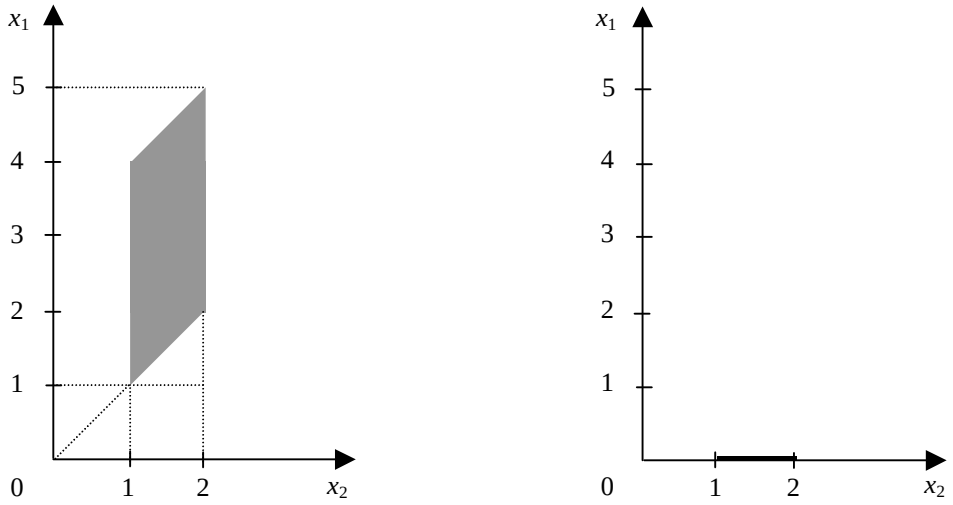
■

Les successeurs continus et discrets d'une région donnée, (L_m, Q_n), définissent l'ensemble des états atteignables à partir d'un état $(L_m,\ v) \in (L_m,\ Q_n)$ par la progression du temps ou en franchissant une transition. Nous utilisons cette méthode de calcul des successeurs lors de la construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel. L'algorithme de passage du RdP T-temporel à l'automate temporisé est introduit dans le chapitre suivant.

Par la suite nous présentons la méthode de calcul des prédécesseurs d'une région.

2.2.3 Calcul des prédécesseurs d'une région

Tout état depuis lequel on peut atteindre un état donné est un prédécesseur de cet état. Il y a deux types de prédécesseurs : continus et discrets.



a. Espace Q_2 b. Espace $Suc_{1,2}$ (Q_2)

FIG. 2.22 – Successeur discret d'une région

Un état depuis lequel on peut atteindre un état donné en laissant le temps s'écouler tout en restant dans le même sommet est un prédécesseur continu de cet état.

Définition 2.13. L'état $(L_m,\ v)$ est un prédécesseur continu de l'état $(L_m,\ v+t)$ si :

$$\exists t \in \mathbb{R}^+ \text{ t.q.} \forall t' \leq t,\ v + t' \models I(L_m)$$
$$(L_m, v) \rightarrow (L_m, v + t).$$

■

La notion de prédécesseur continu est duale à celle de successeur continu.

Tout état depuis lequel on peut atteindre un état donné par le franchissement d'une transition est un prédécesseur discret de cet état.

Définition 2.14. L'état $(L_m,\ v)$ est le prédécesseur discret de l'état $(L_{m+1},\ v')$ par le franchissement de la transition $T_{m,m+1}$=$(L_m,\ a,\ g_{m,m+1},\ A_{m,m+1},\ L_{m+1})$ si :

$$(L_{m,m+1},\ a,\ g_{m,m+1},\ A_{m,m+1},\ L_{m+1}) \in \mathcal{T} \wedge (v \models g_{m,m+1}) \wedge$$
$$\wedge (v' = v[A_{m,m+1}]) \wedge (v[A_{m,m+1}] \models I(L_{m+1}))$$
$$(L_m, v) \rightarrow (L_{m+1}, v')$$

■

La notion de prédécesseur discret est duale à celle de successeur discret.

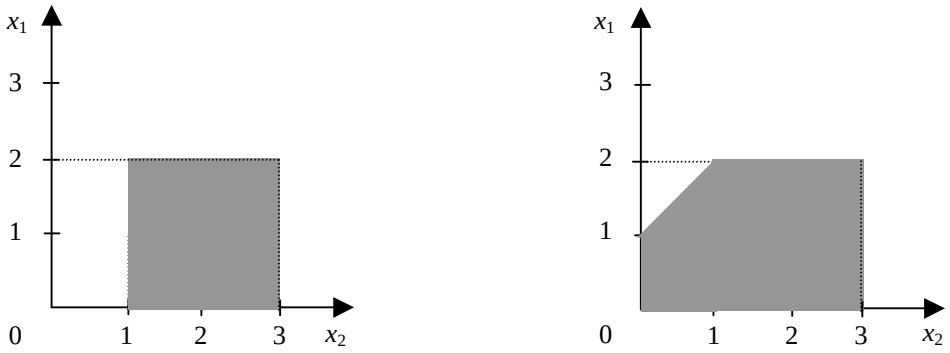
De la même manière que pour les états on peut définir les prédécesseurs d'une région.

Définition 2.15. L'ensemble des états à partir desquels on peut atteindre n'importe quel état $(L_m,\ v) \in (L_m, Q_n)$ en laissant le temps s'écouler, tout en restant dans le même sommet, est appelé prédécesseur continu de la région (L_m, Q_n). Cet ensemble, noté $Pre_t(Q_n)$, est défini par l'expression :

$$v' \models Pre_t(Q_n) \text{ ssi } \exists t \in \mathbb{R}^+ . v' + t \models Q_n \wedge \forall t' \in \mathbb{R}^+ . t' \leq t \Rightarrow v' + t' \models I(L_m)$$

■

Exemple 2.8. Considérons maintenant la région $(L_2,\ Q_3)$ dans le sommet L_2 de l'automate temporisé illustré dans la figure 2.20. L'espace Q_3, représenté dans la figure 2.23.a, est défini par les contraintes suivantes sur la valeur des horloges :



a. Espace Q_3 b. Espace $Pre_t(Q_3)$

FIG. 2.23 – Prédécesseur continu d'une région

$$Q_3 = [0 \leq x_1 \leq 2 \wedge 1 \leq x_2 \leq 3]$$

Le prédécesseur continu de cette région est calculé de la manière suivante :

$$\begin{aligned} Pre_t(Q_3) &= \exists t \in \mathbb{R}^+ . Q_3[x_1 + t, x_2 + t] \\ &= \exists t \in \mathbb{R}^+ . [0 \leq t \wedge 0 \leq x_1 + t \leq 2 \wedge 1 \leq x_2 + t \leq 3] \\ &= [0 \leq x_1 \leq 2 \wedge 0 \leq x_2 \leq 3 \wedge x_2 - x_1 \leq 3 \wedge x_1 - x_2 \leq 1] \end{aligned}$$

Les variables x_1 et x_2 désignent la valeur des horloges dans l'espace $Pre_t(Q_3)$. Selon la définition d'un prédécesseur continu, $(x_1, x_2) \in Pre_t(Q_3)$ ssi $\exists t \in \mathbb{R}^+$ tel que $(x_1 + t, x_2 + t) \in Q_3$. Ainsi, le calcul de l'espace $Pre_t(Q_3)$ est effectué de manière suivante :

- on remplace chaque occurrence d'une variable x_i dans les inégalités qui décrivent l'espace d'horloges Q_3 par $x_i + t$;
- on élimine la variable t.

L'espace d'horloges $Pre_t(Q_3)$ est représenté dans la figure 2.23.b.

■

Définition 2.16. Soit (L_{m+1}, Q_n) une région et $T_{m,m+1}=(L_m, , g_{m,m+1}, A_{m,m+1}, L_{m+1})$ une transition. L'ensemble des états à partir desquels on peut atteindre n'importe quel état $(L_{m+1}, v) \in (L_{m+1}, Q_n)$ en franchissant la transition $T_{m,m+1}$, est appelé prédécesseur discret de la région (L_{m+1}, Q_n). Cet ensemble, noté $Pre_{m,m+1}(Q_n)$, est défini par l'expression :

$$v' \models Pre_{m,m+1}(Q_n) \text{ ssi } v' \models (g_{m,m+1} \wedge I(L_m)) \wedge v'[A_{m,m+1}] \models Q_n$$

■

Exemple 2.9. Considérons la région (L_2, Q_4) dans le sommet L_2 de l'automate temporisé illustré dans la figure 2.20. L'espace Q_4, présenté dans la figure 2.24.b, est défini par les contraintes suivantes sur la valeur des horloges :

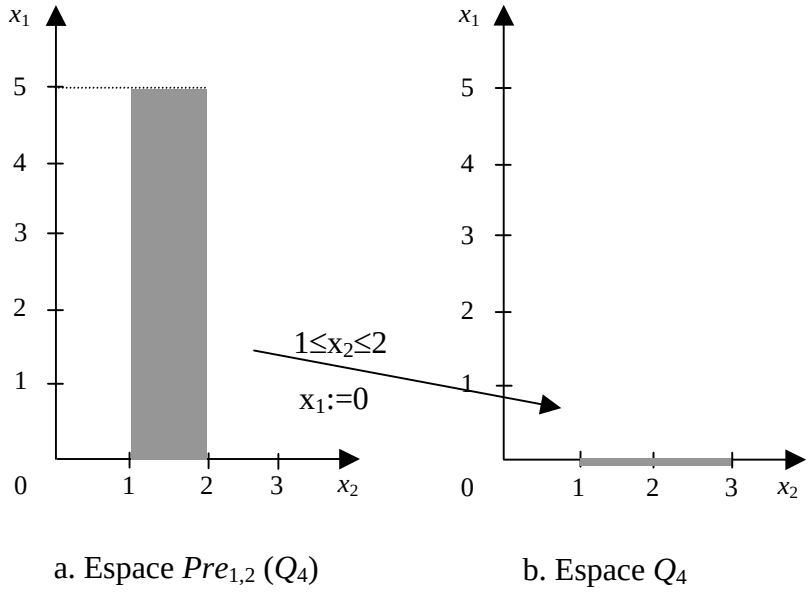
$$Q_4 = [x_1 = 0 \wedge 1 \leq x_2 \leq 3]$$

Par la suite, nous calculons le prédécesseur de cette région par le franchissement de la transition $T_{1,2}=(L_1, , 0 \leq x_2 \leq 2, x_1 := 0, L_2)$. Il faut noter que lors du franchissement de $T_{1,2}$, l'horloge x_1 est mise à zéro par l'affectation associée à cette transition. Le prédécesseur discret de la région (L_2, Q_4) obtenu par le franchissement de $T_{1,2}$ est calculé

de la manière suivante :

$$\begin{aligned} Pre_{1,2}(Q_4) &= Q_4[0, x_2] \wedge g_{1,2} \wedge I(L_1) \\ &= [1 \leq x_2 \leq 3 \wedge 1 \leq x_2 \leq 2 \wedge 0 \leq x_2 \leq 2] \\ &= [1 \leq x_2 \leq 2] \end{aligned}$$

L'espace d'horloges $Pre_{1,2}(Q_4)$ est représenté dans la figure 2.24.a.



a. Espace $Pre_{1,2}$ (Q_4) b. Espace Q_4

FIG. 2.24 – Prédécesseur discret d'une région

■

Nous utilisons ces procédures de calcul des prédécesseurs d'une région lors de la synthèse de la commande, pour déterminer les états à partir desquels on peut atteindre des états interdits. La méthode que nous proposons pour la synthèse de la commande est présentée dans le chapitre 4.

2.3 Conclusion

Dans ce chapitre nous avons présenté les outils de modélisation et analyse qui ont retenu notre attention dans notre travail de recherche. Il s'agit des modèles réseau de Petri(RdP) et automate temporisé.

L'outil RdP est caractérisé par une grande capacité de modélisation. Il permet une représentation intuitive et naturelle des principaux mécanismes des SED : le parallélisme, la synchronisation et le partage des ressources. Le modèle RdP de base est le modèle RdP autonome. Cet outil n'offre que des possibilités d'analyse qualitative. Par conséquent, des extensions ont été définies pour prendre en compte explicitement l'écoulement du temps. Parmi ces extensions, les modèles RdP P-temporels et RdP T-temporels ont retenu notre attention grâce à leur capacité de modéliser la plupart des contraintes temporelles qui interviennent dans le fonctionnement d'un SEDT.

Le modèle RdP P-temporel fournit une représentation naturelle et concise des synchronisations temporelles. Cependant, l'évolution de ce modèle est déterminée par l'occurrence

de deux types d'événements : le franchissement des transitions et la mort des marques dans les places. Ainsi, l'interprétation du modèle est difficile.

La taille du modèle fourni par l'outil RdP T-temporel pour représenter les synchronisations temporelles est plus grande par rapport au modèle RdP P-temporel équivalent. Cependant, l'outil RdP T-temporel offre plus de possibilités pour la modélisation des conflits. De plus, les modèles RdP T-temporels évoluent par l'occurrence d'un seul type d'événements : le franchissement des transitions. Cette caractéristique rend l'interprétation des modèles RdP T-temporels plus simple par rapport aux modèles RdP P-temporels. Ces raisons font que nous avons choisi l'outil RdP T-temporel pour la modélisation du procédé à commander.

Un autre outil de modélisation et analyse des SEDT est le modèle automate temporisé. Les outils basés sur l'automate sont bien appropriés pour l'analyse. En ce qui concerne la modélisation, on peut mentionner l'impossibilité de représenter explicitement certains mécanismes tels que le parallélisme et la synchronisation. De plus, la taille des modèles fournis par ces outils augmente considérablement avec la complexité du système étudié.

L'approche que nous proposons pour la synthèse de la commande par supervision associe la capacité de modélisation de l'outil RdP T-temporel à la puissance d'analyse des automates temporisés. Ainsi, nous représentons le système à commander ainsi que les spécifications imposées sur son fonctionnement par un RdP T-temporel. Généralement, pour analyser les propriétés d'un RdP, on est amené à construire le graphe de marquage qui modélise son comportement. Notre approche propose d'utiliser l'outil automate temporisé pour modéliser le comportement du RdP T-temporel. La technique que nous proposons pour la synthèse de la commande par supervision est basée sur les résultats existants dans la théorie des automates temporisés concernant l'analyse d'atteignabilité des états.

Dans le chapitre suivant nous introduisons l'algorithme que nous proposons pour la construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel.

Chapitre 3

Du RdP T-temporel aux automates temporisés

L'approche de synthèse de la commande par supervision que nous proposons dans ce mémoire associe la capacité de modélisation de l'outil RdP T-temporel à la puissance d'analyse de l'automate temporisé. Par conséquent, une étape importante dans cette approche consiste à construire l'automate temporisé qui modélise le comportement du modèle RdP T-temporel du système à commander.

Dans ce chapitre nous présentons la méthode que nous proposons pour construire d'une manière systématique l'automate temporisé qui modélise le comportement d'un RdP T-temporel donné. D'abord, nous expliquons le principe de cette méthode. Ensuite, nous donnons l'algorithme de construction de l'automate temporisé correspondant à un RdP T-temporel. Enfin, nous précisons la classe des RdP T-temporels qui garantit la convergence de l'algorithme.

3.1 Sur le passage du RdP à l'automate

Une première idée d'associer la capacité de modélisation du modèle RdP à la puissance d'analyse des automates a été matérialisée par une approche d'analyse quantitative du comportement d'un RdP hybride temporisé basée sur l'automate hybride. Cette démarche a été initiée par les travaux de Allam [All98]. Un algorithme a été développé pour construire l'automate hybride qui modélise le comportement d'un RdP hybride temporisé. De plus, une méthode a été proposée pour caractériser le régime périodique d'un RdP hybride en utilisant des techniques d'analyse d'atteignabilité spécifiques aux automates hybrides. Nous avons continué ce travail en apportant des preuves concernant l'unicité de la période déterminé et la convergence de l'algorithme. Les résultats obtenus sont présentés dans [SA01a]. Cette approche est dédiée à l'analyse des systèmes de production modélisés par des RdP hybrides temporisés [AD98a] où la partie continue est un RdP continu à vitesse constante [AD98b] et la partie discrète est un RdP T-temporisé à vitesse maximale [DA92]. Cet outil permet de modéliser des événements qui ont une date d'exécution précise.

Nous rappelons que l'objectif du travail de recherche que nous présentons dans ce mémoire est la synthèse de la commande des SEDT où les événements peuvent avoir lieu dans un intervalle de temps déterminé plutôt qu'à un instant précis. Nous avons donné des arguments pour justifier que l'outil de modélisation le plus adapté à notre démarche est le modèle RdP T-temporel. Ainsi, en s'appuyant sur l'algorithme proposé dans [All98] nous avons développé un nouvel algorithme pour construire l'automate temporisé qui modélise le comportement d'un RdP T-Temporel. La principale difficulté est générée par le non déterminisme sur la date d'occurrence des événements.

D'abord nous avons traité le cas particulier où les variables qui modélisent le passage du temps, i.e. les horloges, sont non couplées [SA] [SAFG01].

Ensuite, nous avons développé l'algorithme dans le cas général, où les horloges sont couplées [SA01b]. La présentation de cet algorithme fait l'objet de ce chapitre.

3.2 Principe du passage du RdP T-temporel aux automates temporisés

Le comportement d'un RdP T-temporel est déterminé par l'évolution de son état suite au franchissement des transitions. Nous rappelons que l'état d'un RdP T-temporel est défini par un couple (M, I), où :

- M est le marquage du réseau. Il caractérise l'état de validation des transitions.
- I est le vecteur des intervalles de franchissement. Il mémorise l'information temporelle concernant le franchissement des transitions validées.

Le marquage des places d'un RdP T-temporel reste inchangé entre deux franchissements successifs. Ainsi, chaque marquage peut être représenté par un sommet d'un automate temporisé. Le franchissement d'une transition d'un RdP T-temporel peut être modélisé par une transition au niveau de l'automate temporisé.

L'information temporelle sur le comportement d'un RdP T-temporel peut être elle-aussi représentée par des éléments d'un automate temporisé :

- Le temps écoulé depuis l'instant de validation d'une transition est mesuré par une horloge.
- L'intervalle de franchissement d'une transition du RdP T-temporel est modélisé par la garde associée à la transition correspondante de l'automate temporisé.

- Le changement du marquage déterminé par le franchissement d'une transition, peut engendrer le changement de l'état de validation des transitions. La mise à zéro des horloges associées aux transitions nouvellement validées est modélisée par l'affectation associée à la transition correspondante de l'automate temporisé.

Toutes ces observations nous ont mené à conclure que le comportement d'un RdP T-temporel peut être modélisé par un automate temporisé.

Pour simplifier l'exposé du principe, nous considérons dans notre démarche que le nombre de validations simultanées des transitions est limité à une.

Généralement, chaque transition d'un RdP T-temporel peut être validée plusieurs fois simultanément. Cela revient à allouer d'une manière dynamique à chaque transition un nombre d'horloges égal au nombre de validations simultanées. Ceci ne concerne que la complexité de l'implémentation. La méthode de construction de l'automate temporisé reste la même.

L'idée de la construction de l'automate temporisé consiste à construire l'ensemble des états atteignables. Cet ensemble est constitué par des sommets correspondant aux marquages discrets. A chaque sommet est associé un espace atteignable correspondant à l'évolution des horloges actives.

Avant de donner l'algorithme que nous proposons pour la construction de l'automate temporisé correspondant à un RdP T-temporel, nous présentons le principe de notre démarche.

3.2.1 Les éléments d'un automate temporisé

Les éléments d'un automate temporisé qui modélise le comportement d'un RdP T-temporel sont identifiés de la manière suivante :

Les horloges : A chaque transition T_j du modèle RdP T-temporel on associe une horloge x_j. Cette horloge sert à compter le temps écoulé depuis l'instant de sa dernière validation. Sa valeur permet de déterminer si, une fois validée, la transition T_j est devenue franchissable. Une horloge est considérée comme étant active lorsque la transition associée est validée.

Les sommets : A chaque marquage M_i du modèle RdP T-temporel on associe un sommet L_i de l'automate temporisé. Des marquages différents sont associés à des sommets différents.

Le changement du marquage d'un RdP T-temporel est déterminé par le franchissement des transitions. Selon les règles de fonctionnement des modèles RdP, une condition nécessaire pour le franchissement d'une transition est qu'elle soit validée par le marquage des places. Par conséquent, pour analyser l'évolution d'un RdP, il suffit de prendre en compte seulement les transitions validées. Pour cette raison, dans chaque sommet de l'automate nous considérons seulement la dynamique des horloges actives.

Les transitions : Le franchissement d'une transition d'un RdP T-temporel peut engendrer un changement de marquage. Cette évolution est modélisée par une transition de l'automate temporisé. L'intervalle de franchissement d'une transition d'un RdP T-temporel mémorise la contrainte temporelle imposée pour son franchissement. Cette contrainte temporelle est modélisée par la garde associée à la transition correspondante de l'automate temporisé. Le changement de marquage dû au franchissement d'une transition peut

déterminer la validation des certaines transitions. L'affectation associée à la transition correspondante de l'automate temporisé met à zéro les horloges associées aux transitions nouvellement validées.

Remarque 3.1. Chaque transition d'un automate temporisé correspond au franchissement d'une seule transition du RdP T-temporel. Par conséquent, sa garde porte toujours sur la valeur d'une seule horloge. L'intervalle de franchissement associé à une transition T_j est $[a_j, b_j]$ ou $[a_j, \infty)$, où a_j et b_j sont des constantes rationnelles. Ainsi, la garde associée à la transition correspondante de l'automate temporisé est définie par l'expression $a_j \prec x_j \prec b_j$ ou $a_j \prec x_j$, $\prec \in \{<, \leq\}$ et x_j est l'horloge associée. Cette propriété caractérise les automates temporisés construits par notre méthode pour modéliser le comportement des RdP T-temporels.

■

Par la suite nous présentons le principe de la méthode de construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel.

3.2.2 Construction de l'automate temporisé

Le sommet initial de l'automate temporisé correspond au marquage initial du RdP. La construction de l'automate temporisé se poursuit en faisant évoluer le modèle RdP T-temporel à partir de son état initial. Le principe de la méthode proposée est illustré dans la figure 3.1.

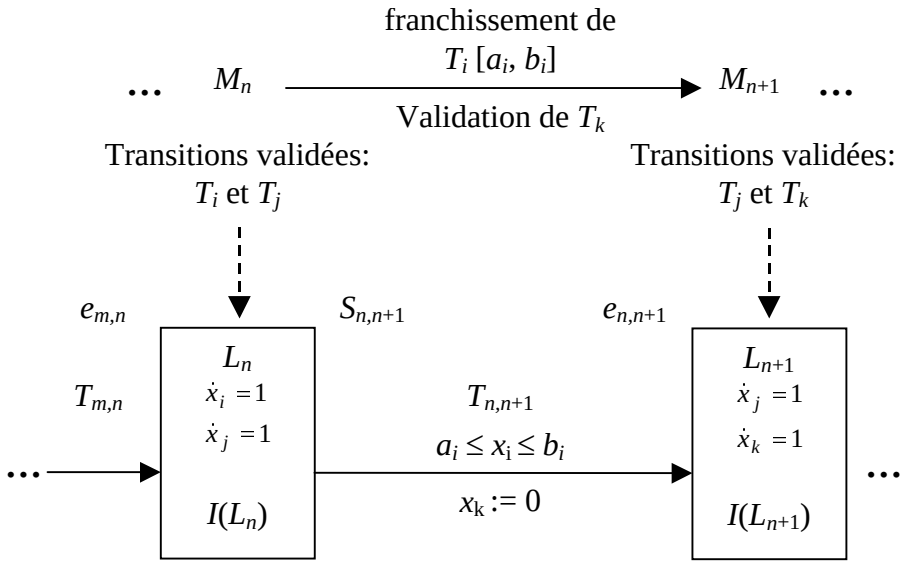


FIG. 3.1 – Schéma de construction d'un automate temporisé

Création d'un sommet

Soit M_n un marquage atteint pendant l'évolution du RdP T-temporel considéré. On modélise ce marquage par le sommet L_n de l'automate temporisé. Supposons maintenant que les transitions validées par ce marquage sont T_i et T_j. A chaque transition T_i du RdP T-temporel on associe une horloge x_i. Ainsi, les horloges actives dans le sommet L_n sont x_i et x_j.

Un système peut séjourner dans un sommet L_n d'un automate temporisé tant que l'invariant $I(L_n)$ associé est vérifié par les valuations des horloges dans ce sommet. Nous rappelons qu'une valuation des horloges est une application qui associe une valeur réelle à chaque horloge. Lorsque le système séjourne dans un sommet, l'ensemble des valuations dans ce sommet définit un espace d'horloge.

Nous avons déjà montré que l'évolution de l'automate temporisé associé à un RdP T-temporel est déterminée seulement par la valeur des horloges actives dans chaque sommet. Ainsi, une connaissance complète de toutes les possibilités d'évolution depuis un sommet L_n implique la connaissance de l'ensemble de toutes valeurs possibles pour les horloges actives dans ce sommet. Par conséquent, pour simplifier le calcul, nous caractérisons l'ensemble des valuations des horloges dans un sommet seulement en fonction des variables associées aux horloges actives dans ce sommet. L'espace défini par cette représentation est appelé espace des horloges actives dans le sommet L_n. Le calcul de cet espace nécessite la connaissance de l'invariant du sommet. Par conséquent, avant de montrer le principe de calcul de l'espace des horloges actives dans un sommet, nous expliquons la méthode de calcul de l'invariant d'un sommet.

Calcul de l'invariant d'un sommet

L'invariant d'un sommet est déduit automatiquement des intervalles de franchissement (conditions de franchissement) des transitions validées par le marquage associé au sommet.

Considérons à nouveau le schéma présenté dans la figure 3.1. L_n est le sommet associé au marquage M_n. Ce marquage permet de valider les transitions T_i et T_j. L'invariant du sommet L_n est déduit automatiquement des intervalles de franchissement de T_i et T_j .

Considérons que l'intervalle de franchissement de la transition T_i est [3, 5]. Cette transition doit être franchie au plus tard lorsque 5 u.t se sont écoulées depuis sa validation. Par conséquent, le système peut séjourner jusqu'à ce que l'horloge x_i atteigne la valeur 5.

Considérons aussi que l'intervalle de franchissement de la transition T_j est [1, 2]. Selon le même raisonnement, le système peut séjourner dans L_n au plus tard jusqu'à ce que l'horloge x_j atteigne la valeur 2.

Par conséquent, le système peut séjourner dans le sommet L_n tant que $x_i \leq 5$ et $x_j \leq 2$. L'invariant du sommet L_n est $I(L_n) = [0 \leq x_i \leq 5 \wedge 0 \leq x_j \leq 2]$.

Le calcul de l'espaces d'horloges actives dans un sommet est basé sur les procédures de calcul des successeurs d'une région présentées dans la section 2.2.2.

Calcul de l'espace des horloges actives dans un sommet

Nous expliquons le principe du calcul de l'espace des horloges actives dans un sommet en nous appuyant à nouveau sur le schéma illustré dans la figure 3.1.

Généralement, lorsqu'un système atteint un sommet L_n d'un automate temporisé par le franchissement d'une transition il y a plusieurs valuations possibles pour les horloges. L'ensemble de ces valuations définit l'espace des horloges à l'entrée du sommet L_n par le franchissement de la transition considérée. Considérons, que le sommet L_n a une transition d'entrée $T_{m,n}$. L'espace des horloges à l'entrée dans ce sommet par le franchissement de cette transition est noté $e_{m,n}$. Cet espace mémorise les valeurs que les horloges peuvent prendre lorsque le système atteint le sommet L_n par le franchissement de $T_{m,n}$.

Cependant, l'évolution de l'automate temporisé depuis le sommet L_n est déterminée seulement par la valeur des horloges actives dans ce sommet. Par conséquent, pour simplifier les calculs, nous exprimons l'espace des horloges $e_{m,n}$ seulement en fonction des

variables associées aux horloges actives dans le sommet L_n. Cet espace, noté $e^a_{m,n}$ est obtenu par la projection orthogonale de l'espace $e_{m,n}$ sur les dimensions des horloges actives dans L_n.

$$e^a_{m,n} = Pr_{horloges\ actives}(e_{m,n})$$

L'espace $e^a_{m,n}$, mémorise les valeurs que les horloges actives dans le sommet L_n peuvent prendre lorsque le système atteint ce sommet. Cependant, cet espace ne garde aucune information sur la valeur des horloges qui ne sont pas actives dans le sommet L_n. Ceci n'est pas un inconvénient, car la valeur de ces horloges n'a aucune influence sur l'évolution du système dans ce sommet.

L'espace des horloges actives dans le sommet L_n est le successeur continu de l'espace des horloges actives à l'entrée dans L_n. Cet espace, noté $Suc_t(e^a_{m,n})$, est calculé par la méthode présentée dans la section 2.2.2.

$$Suc_t(e^a_{m,n}) = \exists t \in \mathbb{R}^+ e^a_{m,n}[x_i - t, x_j - t].I(L_n)$$

Nous rappelons qu'une transition d'un RdP T-temporel peut être franchie seulement si les deux conditions suivantes sont satisfaites :

- elle est validée par le marquage du RdP ;
- sa condition de franchissement, spécifiée par l'intervalle de franchissement associé, est satisfaite.

Il se peut qu'il y ait des transitions du RdP T-temporel qui soient validées par le marquage M_n et dont la condition de franchissement ne soit vérifiée par aucune des valeurs possibles pour les horloges actives dans le sommet associé L_n. Pour éviter d'alourdir inutilement la structure de l'automate temporisé, on ne prend pas en compte le franchissement de ces transitions depuis le sommet considéré L_n. Nous montrons par la suite comment déterminer si le franchissement d'une transition d'un RdP T-temporel doit être pris en compte ou non dans un sommet de l'automate temporisé correspondant.

Déterminer si une transition est franchissable depuis un sommet

Considérons la transition T_i avec l'intervalle de franchissement $[a_i, b_i]$. Cette transition est validée par le marquage M_n, qui est modélisé par le sommet L_n de l'automate temporisé. Le franchissement de cette transition doit être pris en compte dans le sommet L_n s'il y a au moins une valeur possible pour les horloges actives dans L_n qui vérifie sa condition de franchissement. Cette condition est modélisée par l'expression suivante :

$$Suc_t(e^a_{m,n}) \wedge [a_i \leq x_i \leq b_i] \neq \phi$$

Si cette condition est vérifiée, alors il faut créer une transition de sortie du sommet L_n qui modélise son franchissement. Par contre, si elle n'est pas vérifiée, alors le franchissement de la transition T_i n'est pas possible pendant le séjour du système dans le sommet L_n. Dans ce cas nous ne créons pas de transition de sortie du sommet L_n associée au franchissement de cette transition. Nous construisons ainsi l'automate atteignable.

Création d'une transition

Considérons maintenant que le franchissement de la transition T_i est possible pendant le séjour du système dans le sommet L_n. Soit M_{n+1} le marquage atteint par le franchissement de la transition T_i depuis le marquage M_n. Au niveau de l'automate temporisé on crée un nouveau sommet L_{n+1} qui modélise ce marquage. Le franchissement de la transition T_i est modélisé par une transition $T_{n,n+1}$ qui relie les sommets L_n et L_{n+1}.

L'intervalle de franchissement de T_i est $[a_i, b_i]$, donc cette transition est franchie lorsque a_i à b_i u.t. se sont écoulées depuis son instant de validation. Par conséquent, la garde de la transition $T_{n,n+1}$ de l'automate temporisé est $g_{n,n+1} = [a_i \leq x_i \leq b_i]$.

Considérons que les transitions validées par M_{n+1} sont T_j et T_k. Le changement de marquage dû au franchissement de la transition T_i a engendré la validation de T_k. Ainsi, lors du franchissement de la transition $T_{n,n+1}$ il faut mettre à zéro l'horloge associée à la transition T_k, afin de mesurer le temps écoulé depuis sa validation. L'affectation associée à la transition $T_{n,n+1}$ est $A_{n,n+1} = [x_k := 0]$.

Lorsqu'un marquage M_n permet la validation de plusieurs transitions, alors il peut y avoir plusieurs transitions de sortie du sommet associé L_n, donc plusieurs possibilités d'évolution. Les possibilités d'évolution depuis un sommet sont mémorisées dans une pile afin de les analyser ultérieurement une par une.

Calcul de l'espace d'horloges à l'entrée dans un sommet

Nous montrons le principe de calcul de l'espace des horloges à l'entrée dans un sommet en nous appuyant toujours sur le schéma illustré dans la figure 3.1.

En général, la garde d'une transition de sortie d'un sommet n'est pas vérifiée par toutes les valeurs que les horloges actives peuvent prendre dans ce sommet. Soit $T_{n,n+1}$ une transition de sortie du sommet L_n. L'espace des horloges actives dans ce sommet est $Suc_t(e^a_{m,n})$. L'ensemble des valuations des horloges dans le sommet L_n qui vérifient la garde $g_{n,n+1}$ de la transition $T_{n,n+1}$, noté $S_{n,n+1}$, est calculé par la relation suivante :

$$S_{n,n+1} = Suc_t(e^a_{n,n+1}) \wedge g_{n,n+1}$$

L'espace des horloges à l'entrée dans le sommet L_{n+1} par le franchissement de la transition $T_{n,n+1}$, noté $e_{n,n+1}$, est le successeur discret de l'espace $S_{n,n+1}$.

$$e_{n,n+1} = Suc_{n,n+1}(S_{n,n+1})$$

Cet espace est obtenu à partir de l'espace $S_{n,n+1}$ en mettant à zéro les horloges spécifiées par l'affectation associée à la transition $T_{n,n+1}$. Il décrit les valeurs que les horloges peuvent prendre lorsque le système atteint le sommet L_{n+1} par le franchissement de cette transition pendant la visite courante de ce sommet.

Cependant, lors de l'évolution de l'automate temporisé il est possible de visiter plusieurs fois le même sommet, par le franchissement de la même transition, mais avec des espaces des horloges différents à son entrée. Cette situation aparaît lorsque la structure de l'automate contient des boucles. L'ensemble de toutes les valeurs que les horloges peuvent prendre à l'entrée dans un sommet L_{n+1} par le franchissement d'une transition $T_{n,n+1}$ pendant l'évolution de l'automate temporisé est noté $E_{n,n+1}$. Cet espace est l'union des espaces des horloges $e_{n,n+1}$ calculés pour chaque visite du sommet L_n par le franchissement de la transition $T_{n,n+1}$.

Remarque 3.2. L'espace $e^a_{m,n}$ des horloges actives à l'entrée dans un sommet L_n mémorise seulement les valeurs possibles pour les horloges actives à l'entrée dans ce sommet par le franchissement de la transition $T_{m,n}$. En plus par rapport à cet espace, l'espace $e_{m,n}$ mémorise aussi la relation entre les horloges actives dans ce sommet et celles qui ont été désactivées par le franchissement de la transition $T_{m,n}$. Cette information n'est pas utilisée lors de la construction de l'automate temporisé. Cependant, elle est nécessaire pour la synthèse de la commande. Par conséquent, lorsqu'on construit l'automate temporisé, on doit mémoriser l'espace des horloges à l'entrée dans chaque sommet.

■

Remarque 3.3. Les évolutions non-désirées d'un modèle RdP T-temporel sont modélisées par le franchissement de certaines transitions. Un sommet de l'automate temporisé atteint par le franchissement d'une transition non-désiré est appelé interdit. Le but de la commande par supervision est de garantir que les sommets interdits ne sont jamais atteints pendant l'évolution de l'automate temporisé. Pour cette raison, nous ne prenons pas en compte les possibilités d'évolution de l'automate temporisé depuis un sommet interdit.

■

Par la suite, nous présentons l'algorithme que nous proposons pour la construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel donné.

3.3 Algorithme de passage du RdP T-temporels aux automates temporisés

Nous utilisons les notations suivantes :
- $\mathcal{L}$ désigne l'ensemble des sommets ;
- $\mathcal{T}$ est l'ensemble des transitions de l'automate ;
- $\mathcal{M}$ est l'ensemble des marquages du RdP T-temporel ;
- n est un compteur qui mémorise le nombre des sommets ;
- $e_{m,n}$ mémorise l'espace des horloges à l'entrée dans le sommet L_n suite au franchissement d'une transition $T_{m,n}$, pour la visite courante ;
- $E_{m,n}$ est l'espace des horloges à l'entrée dans le sommet L_n, suite au franchissement d'une transition $T_{m,n}$, pour toutes les visites ;
- $S_{m,n}$ dénote l'ensemble des valeurs des horloges dans le sommet L_m qui vérifient la garde de la transition de sortie $T_{m,n}$;
- P est une pile qui mémorise les visites des sommets non encore analysées. Chaque élément de la pile mémorise les caractéristiques d'une visite d'un sommet :
  - le nom du sommet ;
  - les horloges actives dans ce sommet ;
  - l'espace d'horloges à son entrée.

Initialement tous ces ensembles sont vides et le compteur n a la valeur zéro. A chaque transition T_i du RdP T-temporel, on associe une horloge x_i.

Algorithme de passage

Pas 1 : *Initialisation*

Soit M_0 le marquage initial du RdP T-temporel :
- Créer un sommet L_0 associé au marquage M_0 :
  - On détermine les transitions validées par le marquage M_0 ;
  - On détermine les horloges actives dans le sommet L_0. Celles-ci sont les horloges associées aux transitions validées par le marquage M_0.
  - On calcule l'invariant $I(L_0)$ du sommet L_0 à partir des intervalles de franchissement des transitions validées par le marquage M_0.
- Créer une transition d'entrée dans le sommet L_0 et lui associer une affectation qui met à zéro les horloges actives dans L_0 ;
- On détermine l'espace des horloges $e_{0,0}$ à l'entrée du sommet L_0.
  Cet espace contient la valeur 0 pour toutes les horloges actives dans L_0.
- Mémoriser cette visite du sommet L_0 dans la pile P ;

- Actualiser les ensembles :
  - $\mathcal{M} := \{M_0\}$;
  - $\mathcal{L} := \{L_0\}$;
  - $E_{0,0} := e_{0,0}$;
  - $n := 0$.

Pas 2 : *Analyser la dernière visite mémorisée dans la pile*

Supposons qu'il s'agisse de la visite du sommet L_n avec l'espace d'horloges $e_{m,n}$ à son entrée, par le franchissement de la transition $T_{m,n}$

2.1. Enlever de la pile l'élément qui mémorise cette visite, i.e. le dernier élément ;

2.2. Déterminer l'espace des horloges actives dans le sommet L_n :
- On calcule l'espace des horloges actives à son entrée :
$$e^a_{m,n} = Pr_{\text{horloges actives}}(e_{m,n})$$
- On calcule l'espace des horloges actives dans L_n.
Cet espace est le successeur continu de l'espace des horloges actives à l'entrée dans L_n. Il est noté $Suc_t(e^a_{m,n})$.

2.3. Déterminer l'ensemble des transitions du RdP T-temporel qui peuvent être franchies pendant le séjour du système dans le sommet L_n.

2.4. Analyser l'évolution engendrée par le franchissement de chaque transition. Pour chaque transition T_j on détermine le marquage M_{n+1} du RdP T-temporel atteint par son franchissement.

SI $M_{n+1} \notin \mathcal{M}$, alors :
- Créer un sommet L_{n+1} associé au marquage M_{n+1} :
  - On détermine les transitions validées par le marquage M_{n+1} ;
  - On détermine les horloges actives dans le sommet L_{n+1} ;
  - On calcule l'invariant $I(L_{n+1})$ du sommet L_{n+1} à partir des intervalles de franchissement des transitions validées par le marquage M_{n+1}.
- Créer une transition $T_{n,n+1}$=(L_n, , $g_{n,n+1}$, $A_{n,n+1}$, L_{n+1}) qui modélise le franchissement de T_j. On ajoute :
  - sa garde, $g_{n,n+1}$, identique à l'intervalle de franchissement de T_j ;
  - son affectation $A_{n,n+1}$, qui met à zéro les horloges associées aux transitions nouvellement validées.
- Déterminer l'espace des horloges $e_{n,n+1}$ à l'entrée dans L_{n+1} par le franchissement de la transition $T_{n,n+1}$:
  - Calculer l'ensemble des horloges actives dans L_n qui vérifient la garde de $T_{n,n+1}$:
$$S_{n,n+1} = Suc_t(e^a_{m,n}) \wedge g_{n,n+1}.$$
  - Calculer l'espace des horloges $e_{n,n+1}$:
$$e_{n,n+1} = Suc_{n,n+1}(S_{n,n+1}).$$
- Mémoriser dans la pile P cette visite du sommet L_{n+1} avec l'espace $e_{n,n+1}$ à son entrée ;
- Actualiser les ensembles :
  - $\mathcal{M} := \mathcal{M} \cup \{M_{n+1}\}$;
  - $\mathcal{L} := \mathcal{L} \cup \{L_{n+1}\}$;
  - $E_{n,n+1} := e_{n,n+1}$;
  - $n := n + 1$.

SINON Soit $M_l = M_{n+1}$, alors :
- Créer, si elle n'existe pas déjà, une transition $T_{n,l}$=(L_n, , $g_{n,l}$, $A_{n,l}$, L_l) qui modélise le franchissement de T_j. On ajoute :
  - sa garde, $g_{n,l}$, déduite à partir de son intervalle de franchissement,

– son affectation, $A_{n,l}$, qui met à zéro les horloges associées aux transitions nouvellement validées.

- Déterminer l'espace des horloges $e_{n,l}$ à l'entrée dans L_l, par le franchissement de la transition $T_{n,l}$:
  – Calculer l'ensemble des horloges actives dans L_n qui vérifient la garde de $T_{n,l}$:
  $$S_{n,l} = Suc_t(e^a_{m,n}) \wedge g_{n,l}.$$
  – Calculer l'espace des horloges $e_{n,l}$:
  $$e_{n,l} = Suc_{n,l}(S_{n,l}).$$
  SI $e_{n,l} \not\subset E_{n,l}$, alors :
  – Mémorisez dans la pile P cette nouvelle visite du sommet L_l avec l'espace d'horloges $e_{n,l}$ à son entrée ;
  – Actualiser l'ensemble des horloges à l'entrée dans ce sommet :
  $$E_{n,l} := E_{n,l} \vee e_{n,l}$$
  SINON l'évolution de l'automate depuis L_l a déjà été analysée pour ces valeurs des horloges.

Pas 3 : SI $P \neq \phi$, il reste encore des visites des sommets à analyser. *Aller au Pas 2.*

Pas 4 : FIN

■

Par la suite, nous illustrons l'application de cet algorithme à travers un exemple. Nous précisons que notre démarche est générale, mais par simplification, nous considérons dans cet exemple un RdP T-temporel sauf.

Exemple 3.1. Considérons l'exemple du poste de collage présenté dans la section 2.1.5. Le RdP T-temporel qui modélise le fonctionnement de ce processus est illustré dans la figure 2.18. Nous appliquons l'algorithme que nous venons de présenter pour construire l'automate temporisé qui modélise son comportement. Pour améliorer la clarté de la présentation nous reprenons ce RdP T-temporel dans la figure 3.2.

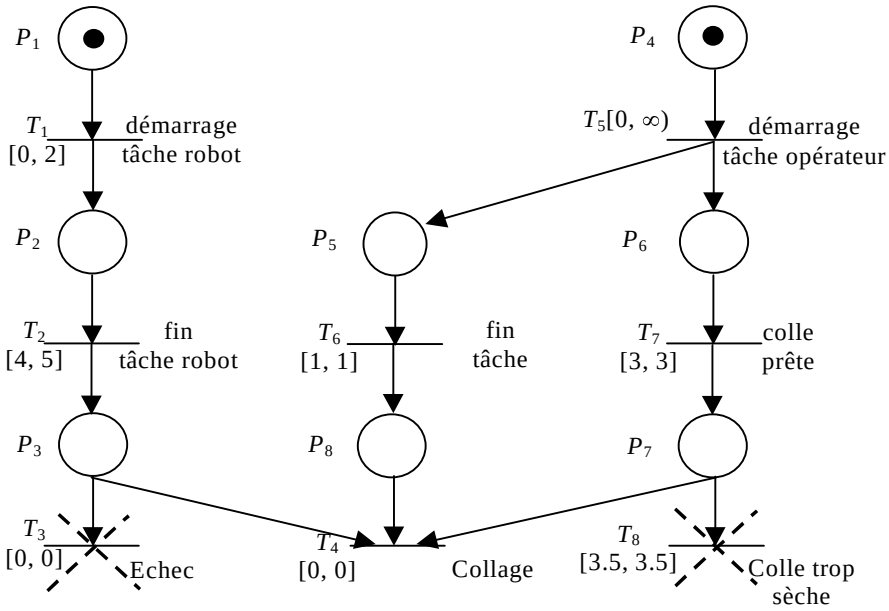


FIG. 3.2 – RdP T-temporel du poste de collage

D'abord, à chaque transition T_i, i=1,...,8, on associe une horloge x_i.

Avant de commencer la construction de l'automate temporisé, nous rappelons que les évolutions non-désirées de ce RdP T-temporel correspondent aux franchissements des transitions T_3 et T_8. Les sommets associées aux marquages atteints par le franchissement de ces transitions sont des sommets interdits. Initialement, tous les ensembles utilisés pour la construction de l'automate temporisé sont vides et le compteur n à la valeur 0.

Pas 1 : *Initialisation*

Le marquage initial du RdP T-temporel est M_0= [1, 0, 0, 1, 0, 0, 0, 0].

- On crée le sommet L_0 associé à M_0. Il est le sommet initial de l'automate temporisé.
  - Les transitions validées par ce marquage sont T_1 et T_5.
  - Les horloges actives dans L_0 sont x_1 et x_5.
  - On calcule l'invariant I(L_0). L'intervalle de franchissement de la transition T_1 est [0, 2] et celui de T_5 est [0, ∞). Ainsi, l'invariant du sommet L_0 est :
$$I(L_0) = [0 \leq x_1 \leq 2].$$
- On crée une transition d'entrée dans L_0 et on lui associe une affectation qui met à zéro la valeur des horloges actives dans ce sommet :
$$A_{0,0}=[x_1 := 0 \wedge x_5 := 0].$$
- L'espace d'horloges à l'entrée dans L_0 est :
$$e_{0,0} = [x_1 = 0 \wedge x_5 = 0].$$
- On mémorise dans la pile P la visite de L_0 avec l'espace d'horloges $e_{0,0}$ à son entrée :
$$P := \{(L_0; x_1, x_5; e_{0,0} = [x_1 = 0 \wedge x_5 = 0])\}$$
- On actualise les ensembles :
  - $\mathcal{M}$:=$\{M_0\}$;
  - $\mathcal{L}$:=$\{L_0\}$;
  - $E_{0,0}$:=$e_{0,0}$;
  - $n := 0$.

La partie de l'automate temporisé construite pendant le premier pas (l'étape d'initialisation de l'algorithme) est présentée dans la figure 3.3.

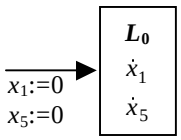


FIG. 3.3 – Automate temporisé obtenu après le premier pas

Pas 2 : (première itération) On analyse la visite du sommet L_0 avec l'espace d'horloges $e_{0,0} = [x_1 = 0 \wedge x_5 = 0]$ à son entrée. Les horloges actives dans L_0 sont x_1 et x_5.

2.1. On enlève l'élément correspondant de la pile :
$$P := \phi.$$

2.2. Déterminer l'espace des horloges actives dans le sommet L_0 :
- On calcule l'espace des horloges actives à l'entrée dans L_0 :

$$\begin{aligned} e^a_{0,0} &= Pr_{x_1,x_5}(e_{0,0}) \\ &= [x_1 = 0 \wedge x_5 = 0] \end{aligned}$$

– On calcule l'espace des horloges actives dans L_0. Il est le successeur continu de l'espace $e^a_{0,0}$.

$$\begin{aligned} Suc_t(e^a_{0,0}) &= \exists t \in \mathbb{R}^+ . e^a_{0,0}[x_1 - t, x_5 - t] \wedge I(L_0) \\ &= \exists t \in \mathbb{R}^+ . [x_1 - t = 0 \wedge x_5 - t = 0 \wedge 0 \leq x_1 - t \leq 2] \wedge \\ &\quad \wedge [0 \leq x_1 \leq 2] \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \end{aligned}$$

L'espace $Suc_t(e^a_{0,0})$ est illustrée dans la figure 3.4.

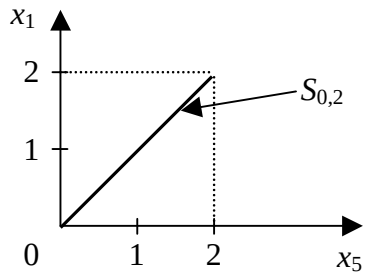


FIG. 3.4 – Successeur continu de l'espace $e^a_{0,0}$

2.3. On détermine si T_1 et/ou T_5 peuvent être franchies pendant cette visite du sommet L_0 :

– *Transition* T_1 :
L'intervalle de franchissement de cette transition est [0, 2]. On teste la possibilité de franchir T_1 pendant cette visite du sommet L_0 :

$$\begin{aligned} Suc_t(e^a_{0,0}) \wedge [0 \leq x_1 \leq 2] &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \wedge [0 \leq x_1 \leq 2] \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \\ &\neq \phi \end{aligned}$$

La transition T_1 peut être franchie pendant cette visite du sommet L_0. Ainsi, il faudra créer une transition de sortie de ce sommet qui modélise le franchissement de cette transition.

– *Transition* T_5 :
L'intervalle de franchissement de cette transition est $[0, \infty)$. On teste la possibilité de franchir T_5 pendant cette visite du sommet L_0 :

$$\begin{aligned} Suc_t(e^a_{0,0}) \wedge [0 \leq x_5] &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \wedge [0 \leq x_5] \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \\ &\neq \phi \end{aligned}$$

La transition T_5 peut être franchie pendant cette visite du sommet L_0. Il faudra créer, également une transition de sortie de ce sommet qui modélise le franchissement de T_5.

A partir de l'état initial, le franchissement de T_1 signifie que le robot démarre sa tâche. Le franchissement de T_5 modélise le démarrage de la tâche de l'opérateur.

2.4. On analyse l'évolution engendrée par le franchissement de chacune des transitions T_1 et T_5.

Analyse du franchissement de T_5

Le marquage du RdP T-temporel atteint par le franchissement de T_5 est M_1=[1, 0, 0, 0, 1, 1, 0, 0].

$M_1 \notin \mathcal{M}$, alors :

- Créer un sommet L_1 associé au marquage M_1 :
  - Les transitions validées par M_1 sont T_1, T_6 et T_7.
  - Les horloges actives dans M_1 sont x_1, x_6 et x_7.
  - On calcule l'invariant du sommet $I(L_1)$ à partir des conditions de franchissement des transitions T_1, T_6 et T_7.
    L'intervalle de franchissement de T_1 est [0, 2]. L'intervalle de franchissement de T_6 est [1, 1] et celui de T_7 est [3, 3]. L'invariant du sommet L_1 est :
    $$I(L_1) = [0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge 0 \leq x_7 \leq 3].$$
- On crée une transition $T_{0,1}$=(L_0, , $g_{0,1}$, $A_{0,1}$, L_1)qui modélise le franchissement de T_5 :
  - L'intervalle de franchissement de T_5 est [0, ∞), donc la garde de $T_{0,1}$ est :
    $$g_{0,1} = [0 \leq x_5]$$
  - Les transitions nouvellement validées suite au franchissement de T_5 sont T_6 et T_7. L'affectation associée à la transition $T_{0,1}$ de l'automate temporisé met à zéro les horloges associées à ces transitions :
    $$A_{0,1} = [x_6 := 0 \wedge x_7 := 0]$$
- On détermine l'espace des horloges $e_{0,1}$ à l'entrée dans le sommet L_1 par le franchissement de la transition $T_{0,1}$:
  - On calcule l'ensemble des horloges actives dans L_0 qui vérifient la garde de $T_{0,1}$:
    $$\begin{aligned} S_{0,1} &= Suc_t(e^a_{0,0}) \wedge g_{0,1} \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \wedge [0 \leq x_5] \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2]. \end{aligned}$$
  - On calcule l'espace des horloges $e_{0,1}$:
    $$\begin{aligned} e_{0,1} &= Suc_{0,1}(S_{0,1}) \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$
- On mémorise dans la pile P la visite de L_1 avec l'espace d'horloges $e_{0,1}$ à son entrée :
  $$P : = \{(L_1; x_1, x_6, x_7; e_{0,1} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0])\}$$
- On actualise les ensembles :
  - $\mathcal{M} = \{M_0, M_1\}$;
  - $\mathcal{L} = \{L_0, L_1\}$;
  - $E_{0,1} = e_{0,1} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0]$;
  - $n := n + 1 = 1$.

Analyse du franchissement de T_1

Le marquage du RdP T-temporel atteint par le franchissement de T_1 est M_2= [0, 1, 0, 1, 0, 0, 0, 0].

$M_2 \notin \mathcal{M}$, alors :

- Créer un sommet L_2 associé au marquage M_2 :
  - Les transitions validées par M_2 sont T_2 et T_5.

- Les horloges actives dans M_2 sont x_2 et x_5.
- On calcule l'invariant du sommet $I(L_2)$ à partir des conditions de franchissement des transitions T_2 et T_5.
  L'intervalle de franchissement de T_2 est [4, 5] et celui de T_5 est [0, ∞). L'invariant du sommet L_2 est :
  $$I(L_2) = [0 \leq x_2 \leq 5].$$

- On crée une transition $T_{0,2}$=(L_0, , $g_{0,2}$, $A_{0,2}$, L_2) qui modélise le franchissement de T_1.
  - L'intervalle de franchissement de T_1 est [0, 2], donc la garde de $T_{0,2}$ est :
    $$g_{0,2} = [0 \leq x_1 \leq 2]$$
  - Le franchissement de T_1 engendre la validation de la transition T_2. L'affectation associée à la transition $T_{0,2}$ de l'automate temporisé met à zéro l'horloge associée à cette transition :
    $$A_{0,2} = [x_2 := 0]$$
- Déterminer l'espace des horloges $e_{0,2}$ à l'entrée dans le sommet L_2 par le franchissement de la transition $T_{0,2}$:
  - On calcule l'ensemble des horloges actives dans L_0 qui vérifient la garde de $T_{0,2}$:
    $$\begin{aligned} S_{0,2} &= Suc_t(e^a_{0,0}) \wedge g_{0,2} \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2] \wedge [0 \leq x_1 \leq 2] \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2]. \end{aligned}$$
  - On calcule l'espace des horloges $e_{0,2}$:
    $$\begin{aligned} e_{0,2} &= Suc_{0,2}(S_{0,2}) \\ &= [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0] \end{aligned}$$

  Cet espace est illustré dans la figure 3.5.

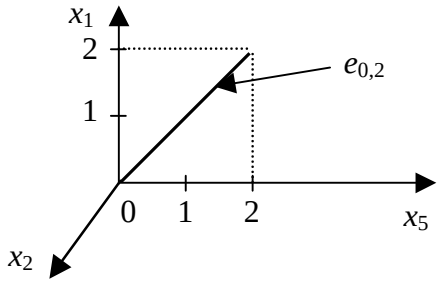


FIG. 3.5 – Espace d'entrée dans L_2 par le franchissement de $T_{0,2}$

- On mémorise dans la pile P la visite de L_2 avec l'espace d'horloges $e_{0,2}$ à son entrée :
  $$\begin{aligned} P \quad := \quad & \{(L_2; x_2, x_5; e_{0,2} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0]), \\ & (L_1; x_1, x_6, x_7; e_{0,1} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0])\} \end{aligned}$$

- On actualise les ensembles :
  - $\mathcal{M} = \{M_0, M_1, M_2\}$;
  - $\mathcal{L} = \{L_0, L_1, L_2\}$;
  - $E_{0,2} = e_{0,2} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0]$;

– $n := n + 1 = 2$.

Pas 3 : $P \neq \phi$, donc il reste encore des visites des sommets à analyser. *Aller au Pas 2.*

L'automate temporisé obtenu à la fin de la première itération de l'algorithme est illustré dans la figure 3.6.

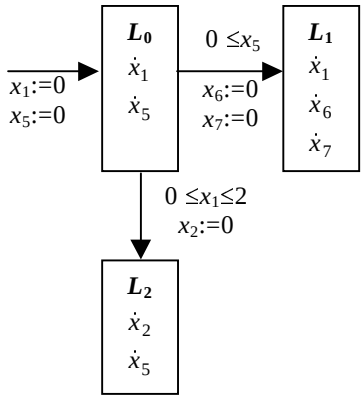


FIG. 3.6 – Automate temporisé obtenu après la première itération

Pas 2 : (deuxième itération) On analyse la visite du sommet L_2 avec l'espace $e_{0,2} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0]$ à son entrée. Les horloges actives dans L_2 sont x_2 et x_5.

2.1. On enlève l'élément correspondant de la pile. Il reste :

$$P \quad := \quad \{(L_1; x_1, x_6, x_7; e_{0,1} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0])\}$$

2.2. Déterminer l'espace des horloges actives dans le sommet L_2 :

– On calcule l'espace des horloges actives à l'entrée dans L_2 :

$$\begin{aligned} e^a_{0,2} &= Pr_{x_2,x_5}(e_{0,2}) \\ &= [0 \leq x_5 \leq 2 \wedge x_2 = 0] \end{aligned}$$

Cet espace est représenté dans la figure 3.7.a.

– On calcule l'espace des horloges actives dans L_2, illustré dans la figure 3.7.b. C'est le successeur continu de l'espace $e^a_{0,2}$.

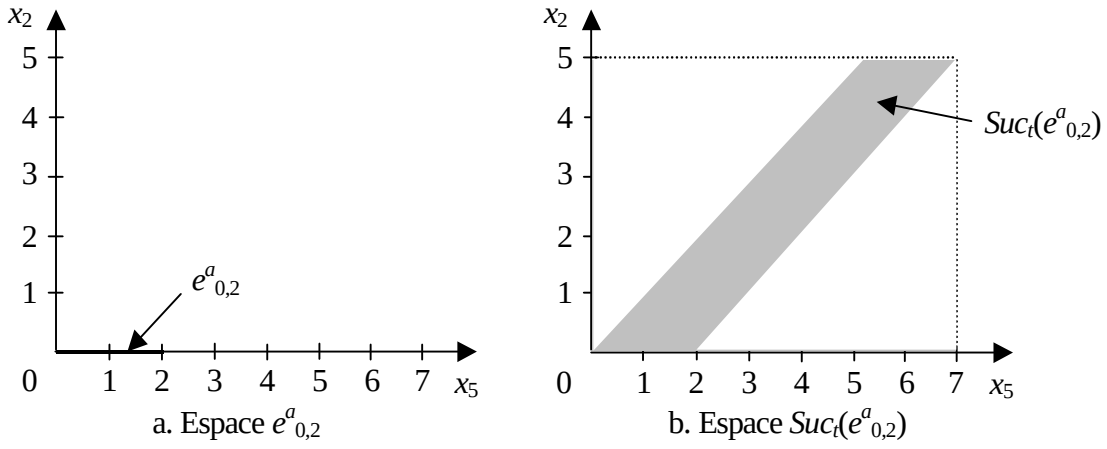


FIG. 3.7 – Espace des horloges dans le sommet L_2

$$\begin{aligned} Suc_t(e^a_{0,2}) &= \exists t \in \mathbb{R}^+ . e^a_{0,2}[x_2 - t, x_5 - t] \wedge I(L_2) \\ &= \exists t \in \mathbb{R}^+ . [0 \leq x_5 - t \leq 2 \wedge x_2 - t = 0] \wedge [0 \leq x_2 \leq 5] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \end{aligned}$$

2.3. On détermine si T_2 et/ou T_5 peuvent être franchies pendant cette visite du sommet L_2 :

– *Transition* T_2 :
L'intervalle de franchissement de cette transition est [4, 5]. On teste la possibilité de franchir T_2 :

$$\begin{aligned} Suc_t(e^a_{0,2}) \wedge [4 \leq x_2 \leq 5] \\ = [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \wedge [4 \leq x_2 \leq 5] \\ = [0 \leq x_5 - x_2 \leq 2 \wedge 4 \leq x_2 \leq 5] \\ \neq \phi \end{aligned}$$

La transition T_2 peut être franchie pendant cette visite du sommet L_2. Ainsi, il faudra créer une transition de sortie de ce sommet qui modélise le franchissement de cette transition.

– *Transition* T_5 :
L'intervalle de franchissement de cette transition est $[0, \infty)$. On teste la possibilité de franchir T_5 :

$$\begin{aligned} Suc_t(e^a_{0,2}) \wedge [0 \leq x_5] \\ = [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \wedge [0 \leq x_5] \\ = [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \\ \neq \phi \end{aligned}$$

La transition T_5 peut être franchie pendant cette visite du sommet L_2. Il faudra créer, également une transition de sortie de ce sommet qui modélise le franchissement de cette transition.

2.4. On analyse l'évolution engendrée par le franchissement de chacune des transitions T_2 et T_5.

Analyse du franchissement de T_2

Le marquage du RdP T-temporel atteint par le franchissement de T_2 est M_3= [0, 0, 1, 1, 0, 0, 0, 0].

$M_3 \notin \mathcal{M}$, alors :

- Créer un sommet L_3 associé au marquage M_3 :

– Les transitions validées par M_3 sont T_3 et T_5.
– Les horloges actives dans L_3 sont x_3 et x_5.
– On calcule l'invariant du sommet $I(L_3)$ à partir des intervalles de franchissement des transitions T_3 et T_5.
L'intervalle de franchissement de T_3 est [0, 0] et celui de T_5 est $[0, \infty)$. L'invariant du sommet L_3 est :

$$I(L_3) = [x_3 = 0].$$

- On crée une transition $T_{2,3}$=(L_2, , $g_{2,3}$, $A_{2,3}$, L_3) qui modélise le franchissement de T_2.

– L'intervalle de franchissement de T_2 est [4, 5], donc la garde de $T_{2,3}$ est :

$$g_{2,3} = [4 \leq x_2 \leq 5].$$

– La transition nouvellement validée suite au franchissement de T_2 est T_3. L'affectation associée à la transition $T_{2,3}$ de l'automate temporisé met à zéro les horloges associées à ces transitions :

$$A_{2,3} = [x_3 := 0]$$

- On détermine l'espace des horloges $e_{2,3}$ à l'entrée dans le sommet L_3 par le franchissement de la transition $T_{2,3}$:
  – On calcule l'ensemble des horloges actives dans L_2 qui vérifient la garde de $T_{2,3}$:

$$\begin{aligned} S_{2,3} &= Suc_t(e^a_{0,2}) \wedge g_{2,3} \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \wedge [4 \leq x_2 \leq 5] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 4 \leq x_2 \leq 5] \end{aligned}$$

L'espace $S_{2,3}$ est représenté dans la figure 3.8.

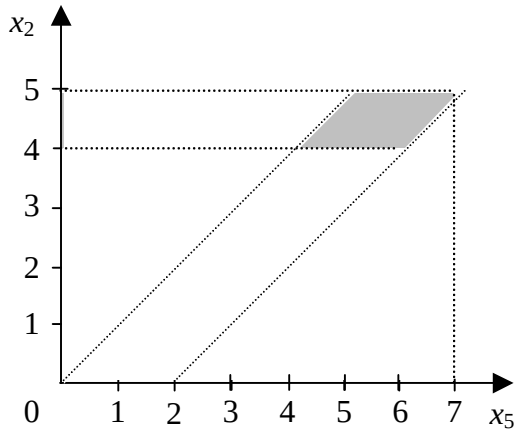


FIG. 3.8 – Espace de sortie du sommet L_2 par le tir de $T_{2,3}$

– On calcule l'espace des horloges $e_{2,3}$:

$$\begin{aligned} e_{2,3} &= Suc_{2,3}(S_{2,3}) \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 4 \leq x_2 \leq 5 \wedge x_3 = 0] \end{aligned}$$

- On mémorise dans la pile P la visite de L_3 avec l'espace d'horloges $e_{2,3}$ à son entrée :

$$\begin{aligned} P \quad := \quad & \{(L_3; x_3, x_5; e_{2,3} = [0 \leq x_5 - x_2 \leq 2 \wedge 4 \leq x_2 \leq 5 \wedge x_3 = 0]) \\ & (L_1; x_1, x_6, x_7; e_{0,1} = [x_1 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge x_6 = 0 \wedge x_7 = 0])\} \end{aligned}$$

– Actualiser les ensembles :
  – $\mathcal{M} = \{M_0, M_1, M_2, M_3\}$;
  – $\mathcal{L} = \{L_0, L_1, L_2, L_3\}$;
  – $E_{2,3} = e_{2,3} = [0 \leq x_5 - x_2 \leq 2 \wedge 4 \leq x_2 \leq 5 \wedge x_3 = 0]$;
  – $n := n + 1 = 3$.

L'automate temporisé obtenu à cet instant de l'exécution de l'algorithme est illustré dans la figure 3.9

En continuant l'exécution de l'algorithme on obtient l'automate temporisé présenté dans la figure 3.10.

Pour faciliter la compréhension du modèle, à chaque sommet nous avons indiqué les places marquées au niveau du RdP T-temporel.

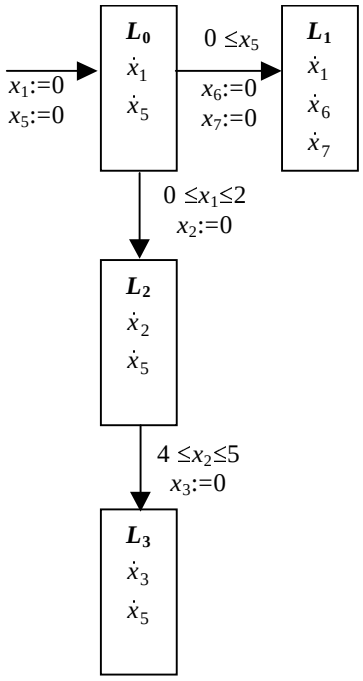


FIG. 3.9 – Automate temporisé partiel

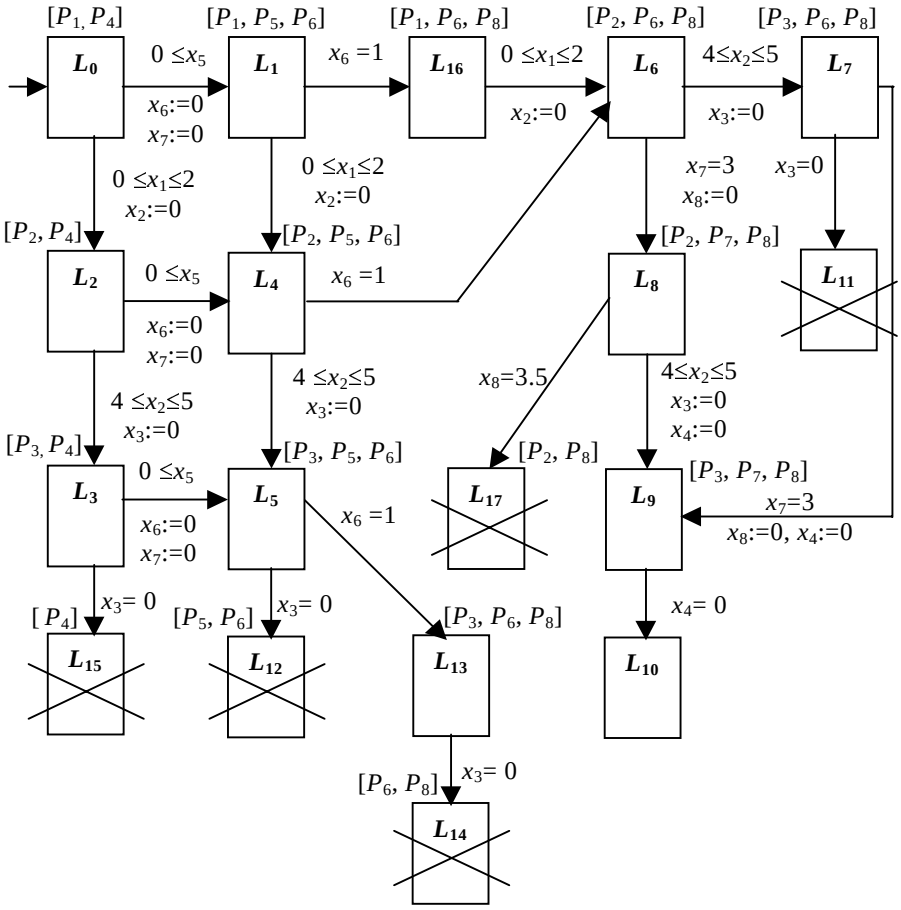


FIG. 3.10 – Automate temporisé associé au poste de collage

Les sommets L_{11}, L_{12}, L_{14}, L_{15} et L_{17} sont des sommets interdits. Ils correspondent aux marquages atteints par le franchissement des transitions T_3 et T_8. Par conséquent, on ne s'intéresse pas aux possibilités d'évolution depuis ces sommets. Le sommet L_{10} modélise la réalisation d'un collage réussi.

3.4 Analyse de l'algorithme

Proposition 3.4.1. *L'algorithme converge pour des RdP T-temporels bornés, avec les intervalles de franchissement spécifiés par des nombres rationnels.*

Preuve: L'algorithme se termine lorsqu'il n'y a aucune nouvelle visite d'un sommet à analyser. Une visite d'un sommet est complètement caractérisée par le marquage du RdP T-temporel et par l'espace des horloges actives à son entrée. Ainsi, l'algorithme converge si le nombre de sommets ainsi que le nombre de fois qu'on visite chaque sommet avec des espaces d'horloges distincts à son entrée est fini.

D'abord, nous montrons que l'automate temporisé associé à un RdP T-temporel dans la classe considérée a un nombre fini de sommets.

Un RdP T-temporel borné est caractérisé par le fait que le nombre de marques dans chaque place est fini. Par conséquent, le nombre de vecteurs de marquage est fini. Chaque vecteur de marquage atteint pendant l'évolution du RdP T-temporel est représenté par un sommet de l'automate temporisé. Ainsi, l'automate temporisé associé à un RdP T-temporel dans la classe considérée a un nombre fini de sommets.

Analysons maintenant le nombre des espaces d'horloges distincts possibles à l'entrée dans un sommet. Par simplification, nous supposons que les intervalles de franchissement associés aux transitions sont spécifiés par des nombres entiers. Cette hypothèse n'est pas restrictive. En effet, lorsque les intervalles de franchissement sont spécifiés par des nombres rationnels, on peut se ramener facilement aux nombres entiers par multiplication avec le plus petit multiple commun de dénominateurs.

Les intervalles de franchissement des transitions d'un RdP T-temporel sont modélisés par les gardes associées aux transitions de l'automate temporisé. Lorsque les intervalles de franchissement sont délimités par des nombres entiers, alors au niveau des gardes, les horloges sont comparées toujours avec des nombres entiers. Ainsi, les contraintes qui décrivent l'espace d'horloges à l'entrée dans un sommet sont toujours spécifiés par des nombres entiers.

Dans [Yov98] on montre que l'espace de la valeur des horloges peut être partagé en un nombre fini de régions d'horloges disjointes.

Chaque espace d'horloges à l'entrée dans un sommet peut être décrit par l'union d'un nombre fini de régions d'horloges disjointes [Yov98]. Par conséquent, le nombre des espaces d'horloges distincts à l'entrée dans un sommet est fini. Le nombre de fois qu'on peut visiter un sommet avec des espaces d'horloges distincts à son entrée est fini. Ainsi, l'algorithme termine dans un nombre fini de pas.

■

L'algorithme que nous avons proposé pour construire de l'automate temporisé qui modélise le comportement d'un RdP T-temporel a été développé en concordance avec notre objectif, qui est la synthèse de la commande par supervision. L'automate temporisé obtenu ne modélise pas l'évolution du système depuis les sommets interdits. Sa structure est minimale dans le sens où dans chaque sommet on prend en compte seulement les transitions de sortie qui peuvent être franchies. Cependant, elle dépend du marquage

initial du RdP T-temporel analysé. Généralement, pour le même RdP T-temporel, des automates temporisés différents sont obtenus pour des marquages initiaux différents.

Toutes les variables de l'automate temporisé obtenu avec l'algorithme proposé sont des horloges. De plus, les gardes associées à ses transitions sont définies par des expressions de type $a \prec x_i \prec b$ où x_i est une horloge, $\prec \in \{<, \leq\}$ et a, b sont des constantes rationnelles. L'analyse d'atteignabilité est décidable pour cette classe d'automates temporisés [HKPV95] [PV94] .

3.5 Conclusions

Notre approche de synthèse de la commande par supervision des SEDT propose la modélisation du système à commander par un RdP T-temporel. Généralement, l'analyse des propriétés d'un RdP passe par la construction du graphe de marquage qui modélise son comportement. Cependant, de nombreux résultats sur l'analyse d'atteignabilité des états existent dans la théorie des automates temporisés. Ces arguments nous ont amené à construire l'automate temporisé à partir du RdP T-temporel.

Ce chapitre nous a permis d'introduire l'algorithme que nous proposons pour construire l'automate temporisé qui modélise le comportement d'un RdP T-temporel. La convergence de cet algorithme est garantie pour des RdP T-temporels bornés, avec des intervalles de franchissement associés aux transitions spécifiés par des nombres rationnels. L'automate temporisé obtenu a une structure minimale dans le sens où il représente strictement les évolutions possibles depuis chaque sommet. De plus, il fournit un modèle sans incohérence temporelle. Cependant, il est dédié surtout à la synthèse de la commande car les évolutions possibles depuis les sommets interdits ne sont pas prises en compte.

L'objectif de notre démarche de synthèse de la commande par supervision est de déterminer les nouvelles gardes des transitions de l'automate temporisé telles que les sommets interdits ne soient jamais atteignables. La méthode que nous proposons pour atteindre cet objectif est présentée dans le chapitre 4.

Chapitre 4

Synthèse de la commande

Jusqu'à présent nous avons développé les raisons justifiant notre choix d'utiliser le modèle RdP T-temporel pour représenter les SEDT à commander et le modèle automate temporisé pour la synthèse de la commande. Nous avons proposé un algorithme pour construire l'automate temporisé qui modélise le comportement d'un RdP T-temporel et nous avons montré l'intérêt de cette représentation.

Ce chapitre est consacré à la présentation de la méthode que nous avons développé pour la synthèse de la commande des SEDT.

Dans un premier temps nous proposons une classification des événements qui interviennent dans le fonctionnement d'un SEDT. Ensuite nous focalisons notre attention sur la synthèse de la commande par supervision. La méthode proposée consiste à calculer de nouvelles conditions de franchissement pour les transitions associées aux événements contrôlables telles que les sommets interdits ne soient jamais atteints pendant l'évolution de l'automate.

4.1 Classification des événements

L'évolution d'un SED est caractérisée par l'occurrence d'événements. Cependant, dans la réalité, un événement ne peut pas avoir lieu à n'importe quel instant. Les dates possibles pour l'occurrence d'un événement w_i dans le fonctionnement d'un SEDT sont spécifiées par un intervalle de temps $[a_i, b_i]$, où $a_i,\ b_i \in \mathbb{Q}^+$, t. q. $0 \leq a_i \leq b_i$ et $0 \leq b_i < \infty$. Ces événements sont classifiés en plusieurs catégories selon la possibilité d'intervenir sur la date de leur occurrence. Nous présentons d'abord la classification existante dans la littérature. Ensuite, nous donnons la classification que nous proposons dans le travail présenté dans ce mémoire et qui nous semble plus appropriée pour la commande des SEDT.

4.1.1 Classification existante

Généralement, les travaux sur la commande par supervision existant dans la littérature classifient les événements qui interviennent dans le fonctionnement d'un SEDT en trois catégories : incontrôlables, contrôlables et forçables.

Un événement est appelé incontrôlable si on ne peut pas toujours interdire son occurrence. Considérons un événement w_i dont l'intervalle de temps associé est $[a_i, b_i]$, où $0 \leq a_i \leq b_i < \infty$. L'occurrence de cet événement ne peut plus être interdite lorsqu'on atteint la borne supérieure b_i de l'intervalle de temps associé. Pour cette raison, tout événement dont l'intervalle de temps associé a la borne supérieure finie, est considéré incontrôlable.

Par opposition, on dit qu'un événement est contrôlable si on peut toujours interdire son occurrence. Ainsi, l'intervalle de temps associé à un événement contrôlable a la borne supérieure infinie. Cependant, celle-ci est une condition nécessaire mais pas suffisante. Un événement w_i dont l'intervalle de temps associé est $[a_i, \infty)$ peut être incontrôlable par sa nature.

La troisième catégorie des événements est représentée par les événements forçables. Un événement est appelé forçable s'il peut se produire spontanément ou être forcé par un système extérieur.

Dans les travaux de Brandin et Wonham [BW94] il est dit qu'un événement est contrôlable et forçable si on peut interdire ou forcer son exécution à n'importe quel moment de temps. Cependant, il peut y avoir des événements contrôlables qui ne peuvent pas être forcés. Considérons l'exemple du labyrinthe présenté dans [RW89]. Dans ce labyrinthe se trouvent un chat et une souris. On peut interdire le passage de la souris d'une chambre à l'autre en bloquant la porte, mais on ne peut pas la forcer à franchir la porte. Ainsi, le passage de la souris d'une chambre à une autre est un événement contrôlable mais il n'est pas forçable. A notre sens il y a une ambiguïté dans la définition d'un événement. On a en fait ici deux événements : 1) *fermer la porte*, qui est un événement contrôlable et 2) *franchir la porte* (pour la souris), qui est un événement incontrôlable.

Un événement est considéré incontrôlable et forçable si on peut forcer son exécution, mais on ne peut pas l'interdire indéfiniment. Par exemple, un système antiaérien peut forcer un avion d'atterrir dans 10 minutes, mais il ne peut pas interdire son atterrissage indéfiniment. L'événement atterrissage est incontrôlable, mais on peut forcer son exécution.

Nous proposons ci-dessous une classification qui, à notre sens, est plus simple et plus appropriée à la commande des SEDT.

4.1.2 Classification proposée

Dans nos travaux de recherche sur la commande des SEDT, nous proposons une définition unifiée pour la contrôlabilité des événements. Nous classifions les événements qui interviennent dans le fonctionnement d'un SEDT en deux catégories : contrôlables et incontrôlables. Cette classification, que nous détaillons par la suite, nous semble plus appropriée pour la commande des SEDT.

Dans notre acception, un événement est contrôlable si on peut fixer sa date d'occurrence dans un intervalle de temps donné. L'intervalle de temps associé à un événement contrôlable définit une tolérance sur sa date d'exécution. Par exemple, le démarrage d'une tâche dans un système de production est un événement contrôlable.

Considérons l'exemple du poste de collage présenté dans la sous-section 2.1.5. Le démarrage de la tâche du robot est un événement contrôlable. L'intervalle de temps associé à cet événement est [0, 2]. Cet intervalle de temps modélise la tolérance spécifiée sur la date d'exécution de cet événement. Si on reprend l'exemple de l'avion, l'événement atterrissage est contrôlable parce qu'on peut fixer sa date d'exécution dans un intervalle de temps qui représente, par exemple, la durée maximale de vol.

Un événement contrôlable est toujours forçable.

Par opposition, un événement est appelé incontrôlable si on ne peut pas agir sur sa date d'exécution. L'intervalle de temps associé à un événement incontrôlable modélise l'incertitude sur sa date d'arrivée. Un exemple d'événement incontrôlable dans un système de production est l'accomplissement d'une tâche.

Considérons à nouveau l'exemple du poste de collage. L'événement fin de la tâche du robot est incontrôlable. L'intervalle de temps [4, 5] associé à cet événement décrit l'incertitude sur sa date d'occurrence.

L'objectif de la synthèse de la commande est de déterminer les contraintes sur la date d'exécution des événements contrôlables telles que l'évolution du système respecte les spécifications imposées. Par la suite nous présentons la méthode que nous proposons pour la synthèse de la commande d'un SEDT.

4.2 Synthèse de la commande

La méthode de synthèse de la commande développée dans le travail de recherche présenté dans ce mémoire est basée sur le modèle automate temporisé. Cet outil a été présenté en détail dans le chapitre 2, section 2.2.1.

Le franchissement d'une transition de l'automate modélise l'occurrence d'un événement dans le système à commander. Par conséquent, chaque transition d'un automate temporisé peut être contrôlable ou incontrôlable, selon la nature de l'événement modélisé.

Lors de la construction de l'automate temporisé associé au processus à commander, dans le chapitre 3, pour chaque événement nous avons retenu la contrainte temporelle sur sa date d'arrivée. En modifiant la contrainte temporelle associée à un événement, on peut forcer son exécution à certains instants tels que le fonctionnement du procédé suit la (les) trajectoire(s) désirée(s). Par conséquent, l'approche que nous présentons dans ce mémoire permet une synthèse de commande du point de vue de l'automaticien.

On définit une loi de commande comme étant une application qui associe une date d'exécution à chaque événement contrôlable. Dans un automate temporisé, les dates possibles pour l'exécution d'un événement sont modélisées par la garde associée à la transition correspondante. Par conséquent, un automate temporisé mémorise la succession logique des événements ainsi que les dates possibles pour l'exécution de chaque événement.

Donc, il caractérise l'ensemble de toutes les lois de commande admissibles pour le SEDT modélisé.

L'objectif de notre démarche est de déterminer l'ensemble de toutes les lois de commande qui garantissent que les comportements non désirés du procédé ne sont pas atteints pendant son fonctionnement. Autrement dit, nous nous proposons de calculer l'ensemble de toutes les trajectoires admissibles pour le fonctionnement du procédé. Ensuite, il revient au technologue de choisir parmi ces trajectoires celle qui convient le mieux aux objectifs de la production suivant des critères de performance (production maximale, un certain taux d'utilisation des machines, etc.).

Exemple 4.1. Reprenons l'exemple du poste de collage présenté dans le chapitre 2, section 2.1.5. Supposons que les événements contrôlables qui peuvent avoir lieu dans ce processus sont le démarrage de la tâche du robot et le démarrage de la tâche de l'opérateur. Les autres événements sont incontrôlables. L'objectif de la synthèse de la commande pour ce procédé est illustré dans la figure 4.1. Nous nous intéressons à déterminer les dates de démarrage des tâches du robot et de l'opérateur qui garantissent un collage réussi.

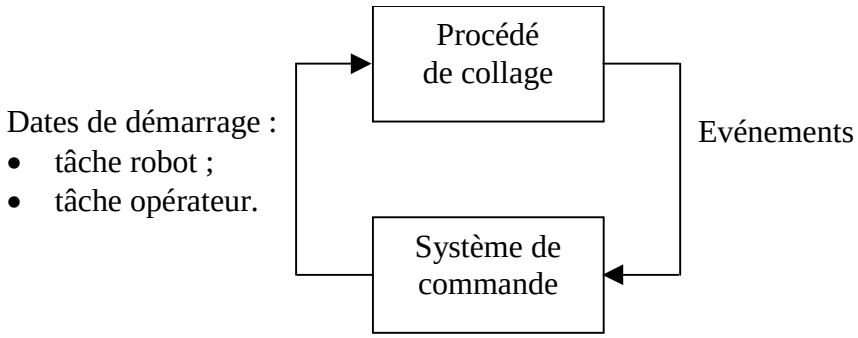


FIG. 4.1 – La synthèse de la commande pour le poste de collage

■

Nous rappelons que les évolutions non désirées d'un procédé sont modélisées par des sommets interdits dans l'automate temporisé associé. La méthode que nous proposons pour la synthèse de la commande est de calculer les nouvelles conditions de franchissement, c'est-à-dire, les gardes des transitions contrôlables telles que les sommets interdits ne soient jamais atteignables.

Les gardes associées aux transitions modélisent des contraintes temporelles imposées sur le fonctionnement du système. On n'a pas les droit d'alléger les contraintes imposées par le cahier des charges. Par conséquent, les contraintes temporelles modélisées par les gardes ne peuvent être que renforcées par rapport à leur valeur initiale, i.e. la largeur de l'intervalle ne peut que diminuer.

De même, l'espace des horloges actives dans un sommet de l'automate temporisé est déterminé par les gardes des transitions en amont du sommet considéré. Donc, lorsqu'on modifie la garde d'une transition de l'automate temporisé, les espaces des horloges actives dans le sommets en aval de cette transition sont réduits par rapport à leur dimension initiale. Ainsi, de la même façon que les gardes, les espaces des horloges actives dans les sommets ne peuvent être que réduits lors de la synthèse de la commande.

Nous rappelons que l'automate temporisé construit en utilisant l'algorithme introduit dans le chapitre précédent a une structure particulière. Il est caractérisé par le fait que la garde de chaque transition porte sur la valeur d'une seule horloge. Elle est définie par

une expression de type $a_j \prec x_j \prec b_j$ ou $a_j \prec x_j$ avec $\prec \in \{<, \leq\}$, x_j est l'horloge associée, et a_j, b_j sont des nombres rationnels.

Remarque 4.1. La garde d'une transition incontrôlable ne peut pas être modifiée lors de la synthèse de la commande. Ainsi, elle est toujours décrite par une expression qui porte sur la valeur d'une seule horloge. Par contre, les gardes des transitions contrôlables ne conservent plus cette propriété.

■

Dans un premier temps, nous expliquons la méthode de calcul des nouvelles gardes associées aux transitions contrôlables telles qu'un sommet interdit n'est jamais atteint pendant l'évolution de l'automate. Ensuite nous donnons l'algorithme pour le cas général.

4.2.1 Principe de synthèse de la commande pour un seul sommet interdit

Considérons la partie d'automate temporisé présentée dans la figure 4.2.

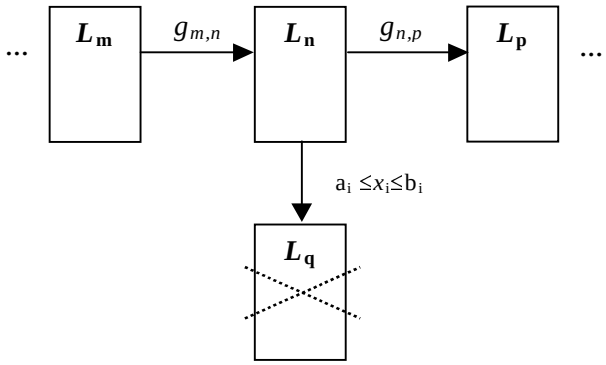


FIG. 4.2 – Partie d'un automate temporisé

Le sommet interdit L_q est atteint depuis le sommet L_n par le franchissement de la transition $T_{n,q}$. Pour qu'on puisse envisager de trouver une solution pour éviter l'évolution vers le sommet interdit L_q, il faut que le sommet L_n ait au moins une autre transition de sortie vers un sommet qui n'est pas interdit. Si cette condition n'est pas vérifiée, alors la seule évolution possible depuis le sommet L_n est de franchir la transition $T_{n,q}$ vers le sommet interdit L_q. Dans ce cas, le sommet L_n devient lui même un sommet interdit.

Soit $T_{n,p}$ une transition de sortie du sommet L_n qui permet au système d'évoluer vers un sommet non interdit L_p. Notre objectif est de calculer les gardes des transitions contrôlables telles que le système soit forcé à quitter le sommet L_n par le franchissement de la transition $T_{n,p}$ avant que la garde de la transition $T_{n,q}$ soit vérifiée par la valeur des horloges dans ce sommet. Ceci revient à déterminer l'ensemble des lois de commande qui ont la propriété que la transition $T_{n,p}$ est toujours franchie avant la transition $T_{n,q}$.

L'espace des horloges actives dans le sommet L_n qui satisfait la propriété que la transition $T_{n,p}$ est toujours franchie avant la transition $T_{n,q}$ est appelé espace des horloges actives désiré dans le sommet L_n. Il est noté D_n.

Généralement, dans la pratique, un comportement non-désiré est engendré par l'occurrence d'un événement incontrôlable. On considère qu'un sommet interdit est toujours atteint par le franchissement d'une transition incontrôlable. En effet, la situation dans laquelle on peut atteindre un sommet interdit par le franchissement d'une transition

contrôlable correspond au fait que l'opérateur lui même décide de provoquer volontairement une évolution non-désirée. Nous considérons qu'un tel comportement ne se produit pas.

La transition $T_{n,q}$, qui mène vers le sommet interdit L_q, est incontrôlable. Ainsi, on ne peut pas agir sur sa date de franchissement. Sa garde ne peut pas être modifiée lors de la synthèse de la commande. Ainsi, il reste deux possibilités d'action pour forcer le franchissement de la transition $T_{n,p}$ avant celui de $T_{n,q}$:

- déterminer une nouvelle garde pour la transition $T_{n,p}$. Ceci est possible seulement si cette transition est contrôlable.
- déterminer les gardes des transitions contrôlables en amont du sommet L_n telles que toutes les valuations des horloges dans ce sommet soient incluses dans l'espace des horloges actives désiré D_n.

Le calcul des gardes des transitions contrôlables telles que la transition $T_{n,p}$ soit toujours franchie avant $T_{n,q}$ est effectué en trois étapes.

La première étape consiste à effectuer les opérations suivantes :

- si $T_{n,p}$ est contrôlable, on cherche une nouvelle garde pour cette transition telle qu'elle soit franchie toujours avant $T_{n,q}$;
- on calcule l'espace D_n des horloges actives désiré dans le sommet L_n.

Cette étape est appelée ***traitement aval***.

Ensuite, on vérifie si on peut atteindre le sommet L_n avec des valuations des horloges qui n'appartient pas à l'espace D_n. Si tel est le cas, on entame la deuxième étape de la synthèse de la commande. Cette étape, appelé ***traitement amont***, consiste à remonter les branches de l'automate et à calculer des nouvelles gardes pour les transitions contrôlables telles que toutes les valuations des horloges dans le sommet L_n appartiennent à l'espace des horloges désiré dans ce sommet.

Enfin, la modification des gardes de certaines transitions peut rendre certains sommets de l'automate temporisé non-atteignables. Ainsi, après avoir calculé des nouvelles gardes pour certaines transitions contrôlables il faut actualiser l'automate temporisé afin de prendre en compte ces modifications. Celle-ci représente la troisième et la dernière étape de la méthode de calcul de l'ensemble des lois de commande qui garantissent que le sommet L_q n'est pas atteignable depuis le sommet L_n.

Avant de donner l'algorithme de la synthèse de la commande, nous détaillons chacune de ses étapes.

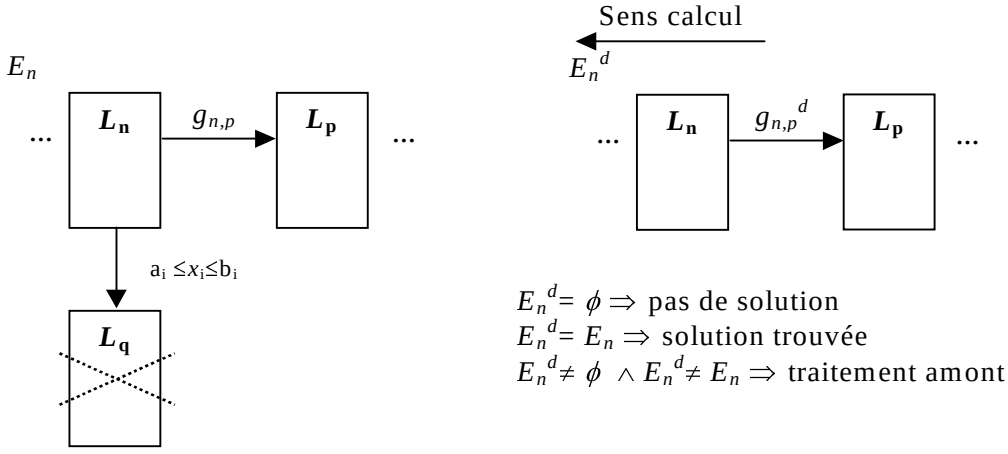
Première étape : traitement aval

Nous avons montré que la garde des transitions de l'automate temporisé construit par l'algorithme que nous avons proposé dans le chapitre 3 a une structure particulière. Elle est décrite par une expression qui porte sur la valeur d'une seule horloge. De plus, la garde d'une transition incontrôlable ne peut pas être modifiée. Alors, elle conserve cette propriété pendant la synthèse de la commande.

La transition $T_{n,q}$, qui mène vers le sommet interdit L_q est incontrôlable. Ainsi, sa garde respecte la propriété qu'on vient de rappeler. Soit $g_{n,q} = [a_i \leq x_i \leq b_i]$ la garde de la transition $T_{n,q}$. Cette transition peut être franchie au plus tôt lorsque l'horloge x_i atteint la valeur a_i.

Par contre, la transition $T_{n,p}$ peut être contrôlable ou incontrôlable, selon la nature de l'événement modélisé. Nous détaillons par la suite chacun de ces deux cas.

Transition $T_{n,p}$ contrôlable Nous expliquons le principe de cette étape dans le cas où la transition $T_{n,p}$ est contrôlable en considérant le schéma illustré dans la figure 4.3.



a. Avant l'étape traitement aval b. Après l'étape traitement aval

FIG. 4.3 – Principe de l'étape traitement aval lorsque $T_{n,p}$ est contrôlable

Initialement, la garde de la transition $T_{n,p}$ est $g_{n,p}$ et l'espace des horloges à l'entrée dans L_n est E_n. Cette situation est illustrée dans la figure 4.3.a.

La transition $T_{n,p}$ est contrôlable, donc sa garde peut être modifiée lors de la synthèse de la commande. Ainsi, les objectifs de l'étape traitement aval sont :

1) Déterminer une nouvelle garde $g^d_{n,p}$, la moins contraignante, de la transition $T_{n,p}$ qui garantit que le système quitte le sommet L_n par le franchissement de $T_{n,p}$ avant que le franchissement de $T_{n,q}$ soit possible ;
2) Calculer l'espace E^d_n des horloges désiré à l'entrée dans le sommet L_n. Cet espace mémorise les valuations des horloges à l'entrée dans L_n qui ne permettent pas une évolution vers le sommet interdit L_q.

La démarche que nous proposons pour atteindre ces objectifs consiste à effectuer les opérations suivantes :

- calculer la nouvelle garde $g^d_{n,p}$ de la transition $T_{n,p}$;
- déterminer l'espace des horloges actives désiré dans le sommet L_n ;
- calculer l'espace E^d_n des horloges désiré à l'entrée dans ce sommet.

La situation obtenue à la fin de cette étape est présentée dans la figure 4.3.b. La transition $T_{n,p}$ a une nouvelle garde qui garantit que le système quitte le sommet L_n par le franchissement de la transition $T_{n,p}$ avant que $T_{n,q}$ puisse être franchie. Ainsi, le sommet L_q n'est plus atteignable. L'espace des horloges désiré à l'entrée dans L_n est E^d_n.

Par la suite, nous détaillons ces opérations.

Calculer la nouvelle garde $g^d_{n,p}$: La nouvelle garde $g^d_{n,p}$ est calculée en ajoutant à l'ancienne garde $g_{n,p}$ la condition que la valeur de l'horloge x_i soit plus petite que a_i.

$$g^d_{n,p} \quad = \quad g_{n,p} \wedge [x_i < a_i] \tag{4.1}$$

Propriété 4.2.1. *La garde $g^d_{n,p}$ calculée par l'équation 4.1 fournit une condition nécessaire et suffisante pour que la transition $T_{n,p}$ soit franchie toujours avant $T_{n,q}$.*

Preuve:

Condition nécessaire : Soit v une valuation des horloges telle que $v \vDash g_{n,p}$ et $a_i \leq v(x_i)$ (ces notations ont été introduites dans le chapitre 2, section 1.2.1).

Lorsque $v \vDash g_{n,p}$, alors la transition $T_{n,p}$ peut être franchie. Cependant, la valeur attribuée à l'horloge x_i par cette valuation satisfait la garde $g_{n,q}$ de la transition $T_{n,q}$. Par conséquent, cette transition est également franchissable. Dans ce cas on ne peut pas empêcher l'évolution du système vers le sommet interdit L_q.

Ainsi, la nouvelle garde $g^d_{n,p} = g_{n,p} \wedge [x_i < a_i]$ fournit une condition nécessaire pour que la transition $T_{n,p}$ soit franchie toujours avant $T_{n,q}$.

Condition suffisante : Soit v une valuation des horloges telle que la garde $g^d_{n,p}$ soit vérifiée, i.e. $v \vDash g^d_{n,p}$.

$$\left.\begin{array}{l} v \vDash g^d_{n,p} \\ g^d_{n,p} = g_{n,p} \wedge [x_i < a_i] \end{array}\right\} \Rightarrow \left\{\begin{array}{l} v \vDash g_{n,p} \\ v(x_i) < a_i \end{array}\right.$$

Lorsque $v \vDash g^d_{n,p}$, alors la transition $T_{n,p}$ peut être franchie. Ce n'est pas le cas de la transition $T_{n,q}$ parce que la valeur attribuée à l'horloge x_i par la valuation v ne permet pas de vérifier la garde de cette transition. Donc, l'évolution vers le sommet interdit L_q n'est pas possible. Par conséquent, la condition fournie par la nouvelle garde $g^d_{n,p}$ est également suffisante.

■

Nous avons trouvé la garde de $T_{n,p}$ la moins contraignante qui garantit que cette transition est toujours franchie avant $T_{n,q}$. Par la suite, il faut vérifier que cette nouvelle garde de $T_{n,p}$ est satisfaite par les valuations des horloges dans le sommet L_n, pour garantir son franchissement.

Calculer l'espace des horloges actives désiré D_n : L'espace des horloges actives désiré dans le sommet L_n mémorise les valuations des horloges dans L_n qui garantissent que le système quitte ce sommet par le franchissement de $T_{n,p}$ avant que la transition $T_{n,q}$ devienne franchissable.

D'abord on calcule l'espace des horloges actives dans le sommet L_n. Le principe de calcul de cet espace a été expliqué dans le chapitre 3, lors de la construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel.

Le calcul de l'espace des horloges actives dans un sommet L_n nécessite la connaissance des éléments suivants :

- l'invariant de ce sommet, $I(L_n)$;
- l'espace des horloges à l'entrée dans le sommet L_n.

L'invariant du sommet L_n est connu. Il a été calculé lors de la construction de l'automate temporisé. Nous montrons plus tard dans ce chapitre que l'invariant d'un sommet peut être également actualisé lors de la synthèse de la commande. Cette opération est nécessaire pour prendre en compte le changement des valeurs maximales que les horloges peuvent prendre dans un sommet. Ce changement est une conséquence de la modification de garde de la transition de sortie du sommet considéré.

De même, les espaces des horloges $E_{m,n}$ obtenus à l'entrée dans L_n par le franchissement de chacune de ses transitions d'entrée $T_{m,n}$ sont également connus. Ils ont été calculés lors de la construction de l'automate temporisé. L'union de ces espaces définit l'espace des horloges à l'entrée dans le sommet L_n, noté E_n.

Nous rappelons que l'évolution d'un automate temporisé dans chaque sommet est déterminée seulement par la valeur des horloges actives. Ainsi, pour simplifier les calculs, nous exprimons l'ensemble des valuations des horloges à l'entrée du sommet L_n en fonction des variables associées aux horloges actives dans ce sommet. Cet espace, noté E^a_n est obtenu

par la projection orthogonale de l'espace E_n sur les dimensions des horloges actives dans le sommet L_n .

$$E_n^a = Pr_{horloges\ actives}(E_n) \tag{4.2}$$

L'espace E_n^a contient toute l'information sur la valeur des horloges actives dans le sommet L_n. Par contre, cet espace ne mémorise aucune information sur la valeur des horloges qui ne sont pas actives dans le sommet L_n. Ceci n'est pas un inconvénient car la valeur de ces horloges n'a aucune influence sur l'évolution du système dans ce sommet .

L'espace des horloges actives dans le sommet L_n est le successeur continu de l'espace E_n^a. Cet espace, noté $Suc_t(E_n^a)$, mémorise toutes les valuations des horloges dans le sommet L_n. Il est calculé par la méthode présentée dans le chapitre 2, section 2.2.2.

Ensuite, on calcule l'ensemble des valuations des horloges qu'on souhaite atteindre dans le sommet L_n.

L'ensemble des valuations qui vérifient la garde $g_{n,p}^d$ noté $S_{n,p}^d$, est défini par l'expression suivante :

$$S_{n,p}^d = Suc_t(E_n^a) \wedge g_{n,p}^d \tag{4.3}$$

L'ensemble des valuations des horloges qui garantissent que la transition $T_{n,p}$ est toujours franchie avant $T_{n,q}$ est formé par deux catégories de valuations :

1) les valuations qui vérifient la nouvelle garde $g_{n,p}^d$ de la transition $T_{n,p}$, i.e. appartiennent à l'espace $S_{n,p}^d$;
2) les valuations à partir desquelles on peut atteindre l'espace $S_{n,p}^d$ en laissant le temps évoluer.

Cet ensemble de valuations est le prédécesseur continu de l'espace $S_{n,p}^d$, noté $Pre_t(S_{n,p}^d)$. Il représente les valuations qu'on souhaite atteindre dans le sommet L_n. La méthode de calcul des prédécesseurs d'un espace des horloges a été présenté dans la section 2.2.2.

L'espace des horloges actives désiré dans le sommet L_n est décrit par l'intersection de l'espace des valuations des horloges existantes dans L_n, noté $Suc_t(E_n^a)$ et de l'espace de valuations qu'on souhaite atteindre dans ce sommet :

$$D_n = Suc_t(E_n^a) \wedge Pre_t(S_{n,p}^d) \tag{4.4}$$

Par la suite, il faut déterminer les valuations des horloges avec lesquelles on peut atteindre le sommet L_n pour que l'évolution vers le sommet interdit L_q ne soit pas possible. L'ensemble des valuations des horloges qui ont cette propriété est mémorisé dans l'espace E_n^d des horloges à l'entrée dans le sommet L_n.

Calcul de l'espace des horloges désiré à l'entrée dans L_n : L'espace des horloges désiré à l'entrée dans le sommet L_n, noté E_n^d, est défini par la relation suivante :

$$E_n^d = E_n \wedge D_n \tag{4.5}$$

Propriété 4.2.2. *L'espace E_n^d définit l'ensemble de toutes les valuations des horloges à l'entrée dans le sommet L_n qui permettent le franchissement de $T_{n,p}$ avec la nouvelle garde.*

Preuve: Soit $v \in E_n$ une valuation des horloges à l'entrée dans L_n telle que $v \notin E_n^d$.

$$\left.\begin{array}{l} v \in E_n \\ v \notin E_n^d \\ E_n^d = E_n \wedge D_n \end{array}\right\} \Rightarrow v \notin D_n$$

Cependant, v est une valuation des horloges dans le sommet L_n. Alors, $v \in Suc_t(E_n^a)$.

$$\left.\begin{array}{l} v \notin D_n \\ v \in Suc_t(E_n^a) \\ D_n = Suc_t(E_n^a) \wedge Pre_t(S_{n,p}^d) \end{array}\right\} \Rightarrow v \notin Pre_t(S_{n,p}^d)$$

Par conséquent, lorsqu'on atteint le sommet L_n avec une valuation $v \notin E_n^d$, alors l'espace $S_{n,p}^d$ ne peut pas être atteint pendant l'évolution du système dans ce sommet. Donc, la nouvelle garde de la transition $T_{n,p}$ n'est plus satisfaite pendant le séjour du système dans ce sommet. Dans ce cas on ne peut plus quitter L_n par le franchissement de la transition $T_{n,p}$ et le système peut évoluer vers le sommet interdit L_q.

Considérons maintenant qu'on atteint le sommet L_n avec une valuation des horloges $v' \in E_n^d$.

$$\left.\begin{array}{l} v' \in E_n^d \\ E_n^d = E_n \wedge D_n \end{array}\right\} \Rightarrow v' \in D_n$$

$$\left.\begin{array}{l} v' \in D_n \\ D_n = Suc_t(E_n^a) \wedge Pre_t(S_{n,p}^d) \end{array}\right\} \Rightarrow v' \in Pre_t(S_{n,p}^d)$$

Par conséquent, lorsqu'on atteint le sommet L_n avec une valuation des horloges $v' \in E_n^d$, l'évolution du système dans ce sommet permet d'atteindre l'espace $S_{n,p}^d$. Ainsi, la nouvelle garde de la transition $T_{n,p}$ est satisfaite et on quitte ce sommet par le franchissement de cette transition avant que l'évolution vers le sommet interdit L_n soit possible.

■

On peut distinguer trois cas, selon la valeur de l'espace E_n^d :

1. $\mathbf{E_n^d = \phi}$: Alors il n'y a aucune valuation des horloges à l'entrée dans le sommet L_n qui garantit que la transition $T_{n,p}$ est toujours franchie avant $T_{n,q}$. Nous rappelons que l'espace des horloges actives dans un sommet de l'automate temporisé ne peut être que réduit lors de la synthèse de la commande. Par conséquent, dans ce cas, il n'y a aucune solution pour forcer le franchissement de la transition $T_{n,p}$ avant celui de $T_{n,q}$. Donc, on ne peut pas éviter l'évolution vers le sommet interdit L_q en forçant le système à quitter le sommet L_n par le franchissement de $T_{n,p}$. Lorsque $T_{n,p}$ est la seule transition de sortie du sommet L_n vers un sommet non-interdit, alors L_n devient lui-même un sommet interdit.
2. $\mathbf{E_n^d = E_n}$: Dans ce cas il n'est pas possible d'atteindre le sommet L_n avec des valuations des horloges qui permettent l'évolution du système vers le sommet interdit L_q. On peut conclure qu'on a trouvé la solution pour garantir que le système quitte le sommet L_n par le franchissement de $T_{n,p}$ avant que l'évolution vers le sommet interdit L_q soit possible. Ainsi, la nouvelle garde que nous avons calculé pour la transition $T_{n,p}$ garantit que le sommet interdit L_q n'est plus atteignable depuis le sommet L_n.
3. $\mathbf{E_n^d \neq \phi}$ et $\mathbf{E_n^d \neq E_n}$: dans ce cas il faut remonter les branches de l'automate et calculer des nouvelles gardes pour les transitions contrôlables telles que toutes les valuations des horloges à l'entrée dans le sommet L_n appartiennent à l'espace E_n^d. Ce traitement est effectué pendant la deuxième étape de la synthèse de la commande, appelée ***traitement amont***. On mémorise dans la pile P le sommet L_n avec l'espace des horloges E_n^d désiré à son entrée.

   De même, les modifications effectuées lors de la synthèse de la commande peuvent faire en sorte que certains éléments de l'automate temporisé situés en amont de

L_n ne soient plus atteignables. Il faut donc actualiser la structure de l'automate temporisé pour prendre en compte ces modifications. Cette opération sera effectuée lors de la troisième étape de la synthèse de la commande. Cependant, pour optimiser cette opération, on garde la trace de ces modifications. On utilise un ensemble Q qui garde les sommets à partir desquels il faut actualiser l'automate temporisé. Dans ce cas, on mémorise le sommet L_n dans l'ensemble Q.

La modification de la garde de la transition $T_{n,p}$ peut déterminer une nouvelle valeur maximale pour certaines horloges dans le sommet L_n. Il faut alors actualiser l'invariant de L_n pour prendre en compte la nouvelle valeur maximale que l'horloge x_i peut prendre dans ce sommet. Le nouvel invariant du sommet $I(L_n)^d$ est décrit par l'expression suivante :

$$I(L_n)^d = I(L_n) \wedge [x_i < a_i] \tag{4.6}$$

Exemple 4.2. Considérons la partie de l'automate temporisé illustrée dans la figure 4.4.

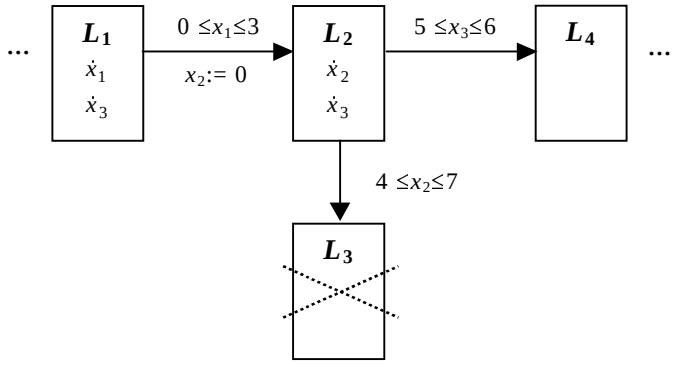


FIG. 4.4 – Partie d'un automate temporisé

Le sommet interdit L_3 est atteint à partir du sommet L_2 par le franchissement de la transition $T_{2,3}$. Cependant, le sommet L_2 a une transition de sortie, $T_{2,4}$, vers le sommet non-interdit L_4. Cette transition est contrôlable. Par conséquent, on peut chercher une nouvelle garde pour cette transition telle qu'elle soit franchie toujours avant $T_{2,3}$.

Calculer la garde $g^d_{2,4}$: La valeur minimale de l'horloge x_2 qui autorise le franchissement de la transition $T_{2,3}$ est 4. Ainsi, la nouvelle garde, $g^d_{2,4}$ de la transition $T_{2,4}$ est calculée par la relation suivante :

$$\begin{aligned} g^d_{2,4} &= g_{2,4} \wedge [x_2 < 4] \\ &= [5 \leq x_3 \leq 6 \wedge x_2 < 4] \end{aligned}$$

Par la suite on vérifie que cette nouvelle garde de la transition $T_{2,4}$ est vérifiée par les valuations des horloges dans le sommet L_2.

Calculer l'espace des horloges actives désiré D_2 : On calcule d'abord l'espace des horloges actives dans le sommet L_2. Cette opération nécessite la connaissance de l'invariant du sommet L_2 et de l'espace des horloges actives à l'entrée dans ce sommet.

Supposons que l'invariant du sommet L_2 est :

$$I(L_2) = [0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6].$$

Par simplification, nous avons considéré que le sommet L_2 a une seule transition d'entrée, $T_{1,2}$. Supposons également que l'espace des horloges $E_{1,2}$ à l'entrée dans ce sommet par le franchissement de $T_{1,2}$ est décrit par l'expression suivante :

$$E_{1,2} = [0 \leq x_1 \leq 3 \wedge x_1 = x_3 \wedge x_2 = 0]$$

L'espace des horloges à l'entrée dans le sommet L_2 est :

$$E_2 = E_{1,2} = [0 \leq x_1 \leq 3 \wedge x_1 = x_3 \wedge x_2 = 0]$$

L'espace des horloges actives à l'entrée dans le sommet L_2 est obtenu par la projection orthogonale de l'espace E_2 sur les dimensions des horloges actives dans ce sommet : x_2 et x_3.

$$\begin{aligned} E_2^a &= Pr_{x_2,x_3}(E_2) \\ &= [0 \leq x_3 \leq 3 \wedge x_2 = 0] \end{aligned}$$

L'espace des horloges actives dans le sommet L_2 est le successeur continu de l'espace E_2^a.

$$\begin{aligned} Suc_t(E_2^a) &= \exists t \in \mathbb{R}^+ . E_2^a[x_2 - t, x_3 - t].I(L_2) \\ &= \exists t \in \mathbb{R}^+ . [0 \leq x_3 - t \leq 3 \wedge x_2 - t = 0] \wedge [0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \\ &= [0 \leq x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \end{aligned}$$

Ensuite, on calcule l'espace $S_{2,4}^d$ qui mémorise les valuations des horloges dans le sommet L_2 qui vérifient la nouvelle garde de la transition $T_{2,4}$. Cet espace est défini par la relation suivante :

$$\begin{aligned} S_{2,4}^d &= Suc_t(E_2^a) \wedge g_{2,4}^d \\ &= [0 \leq x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \wedge [5 \leq x_3 \leq 6 \wedge x_2 < 4] \\ &= [1 < x_3 - x_2 \leq 3 \wedge 5 \leq x_3 \leq 6 \wedge 0 \leq x_2 < 4] \end{aligned}$$

L'ensemble des valuations qui vérifient la garde $(g_{2,4})_n$ ou à partir desquelles on peut atteindre des valuations qui ont cette propriété par l'écoulement du temps est décrit par le prédécesseur continu de l'espace $S_{2,4}^d$. Cet espace, noté $Pre_t(S_{2,4}^d)$ est calculé par la méthode présentée dans la section 2.2.2.

$$\begin{aligned} Pre_t(S_{2,4}^d) &= \exists t \in \mathbb{R}^+ . S_{2,4}^d[x_2 + t, x_3 + t] \\ &= \exists t \in \mathbb{R}^+ . [1 < x_3 - x_2 \leq 3 \wedge 5 \leq x_3 + t \leq 6 \wedge 0 \leq x_2 + t < 4] \\ &= [1 < x_3 - x_2 \leq 3 \wedge 1 < x_3 \leq 6 \wedge 0 \leq x_2 < 4] \end{aligned}$$

L'espace des horloges actives désiré dans le sommet L_2 est défini par l'expression suivante :

$$\begin{aligned} D_2 &= Suc_t(E_2^a) \wedge Pre_t(S_{2,4}^d) \\ &= [0 \leq x_3 - x_2 \leq 3 \wedge 0 \leq x_3 \leq 6 \wedge 0 \leq x_2 \leq 7] \wedge \\ &\quad [1 < x_3 - x_2 \leq 3 \wedge 1 < x_3 \leq 6 \wedge 0 \leq x_2 < 4] \\ &= [1 < x_3 - x_2 \leq 3 \wedge 1 < x_3 \leq 6 \wedge 0 \leq x_2 < 4] \end{aligned}$$

Calculer l'espace des horloges désiré à l'entrée dans L_2 : L'espace des horloges désiré à l'entrée dans le sommet L_2, noté E_2^d, est défini par la relation suivante :

$$\begin{aligned} E_2^d &= E_2 \wedge D_2 \\ &= [0 \leq x_1 \leq 3 \wedge x_1 = x_3 \wedge x_2 = 0] \wedge \\ &\quad [1 < x_3 - x_2 \leq 3 \wedge 1 < x_3 \leq 6 \wedge 0 \leq x_2 < 4] \\ &= [x_1 = x_3 \wedge 1 < x_1 \leq 3 \wedge x_2 = 0] \\ &\neq \phi \end{aligned}$$

En comparant les espaces E_2^d et E_2 on remarque que $E_2^d \neq E_2$. Par conséquent, on peut atteindre le sommet L_2 avec des valuations qui n'appartiennent pas à l'espace D_2.

On mémorise dans une pile P, le sommet L_2 avec l'espace des horloges E_2^d désiré à son entrée. Il faut effectuer un traitement amont que nous verrons plus tard.

De même, il faut actualiser l'automate temporisé en supprimant les éléments qui ont été rendus non-atteignables par la modification de la garde de $T_{2,4}$. Cette opération sera effectuée lors de la troisième étape de la synthèse de la commande, que nous présentons plus tard. On mémorise le sommet L_2 dans l'ensemble Q.

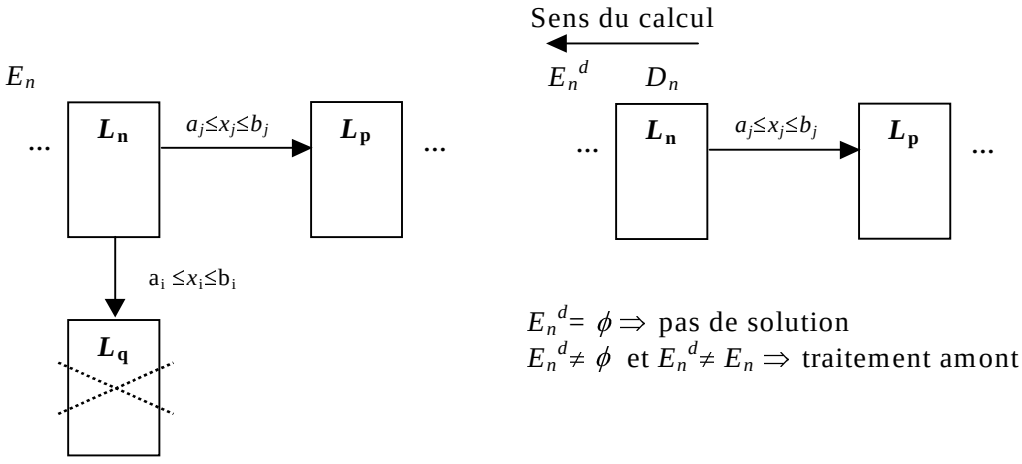
Finalement, on actualise l'invariant du sommet L_2 pour prendre en compte la nouvelle valeur maximale que l'horloge x_2 peut prendre dans le sommet L_2.

$$\begin{aligned} I(L_2)^d &= I(L_2) \wedge [x_2 < 4] \\ &= [0 \leq x_3 \leq 6 \wedge 0 \leq x_2 < 4] \end{aligned}$$

■

Par la suite nous détaillons le cas où la transition $T_{n,p}$ est incontrôlable.

Transition $T_{n,p}$ incontrôlable Nous présentons le déroulement de l'étape traitement aval dans le cas où $T_{n,p}$ est incontrôlable en nous appuyant sur le schéma illustré dans la figure 4.5.



a. Avant l'étape traitement aval b. Après l'étape traitement aval

FIG. 4.5 – Principe de l'étape traitement aval lorsque $T_{n,p}$ est incontrôlable

Initialement, la garde de la transition $T_{n,p}$ est $g_{n,p} = [a_j \leq x_j \leq b_j]$ et l'espace des horloges à l'entrée dans L_n est E_n. Cette situation est illustrée dans la figure 4.5.a.

La transition $T_{n,p}$ est incontrôlable, donc sa garde ne peut pas être modifiée lors de la synthèse de la commande. Ainsi, notre objectif est de calculer l'espace des horloges désiré à l'entrée dans le sommet L_n, tel que la transition $T_{n,p}$ soit franchie toujours avant $T_{n,q}$.

Puisqu'on n'a pas la possibilité de modifier la garde de $T_{n,p}$, on cherche à restreindre l'ensemble des valuations dans L_n à celles qui ne permettent que l'évolution vers le sommet L_p. Par conséquent, dans ce cas, le traitement consiste à calculer l'espace E_n^d des horloges désiré à l'entrée dans L_n.

La démarche que nous proposons pour effectuer ce calcul consiste à effectuer les opérations suivantes :

- calculer la condition sur la valeur des horloges dans le sommet L_n qui garantit que la transition $T_{n,p}$ est toujours franchie avant $T_{n,q}$;
- déterminer l'espace D_n des horloges actives désiré dans L_n ;
- calculer l'espace E_n^d des horloges désiré à l'entrée dans ce sommet.

A la fin de cette étape on obtient l'espace E_n^d des horloges désiré à l'entrée dans le sommet L_n. Cette situation est illustrée dans la figure 4.5.b.

Par la suite, nous détaillons ces opérations.

Calculer la condition sur la valeur des horloges dans L_n : Selon la remarque 4.1, la garde de la transition $T_{n,p}$ porte sur la valeur d'une seule horloge. Soit $g_{n,p} = [a_j \leq x_j \leq b_j]$ la garde de la transition $T_{n,p}$. Ainsi, cette transition doit être franchie au plus tard lorsque l'horloge x_j atteint la valeur b_j.

Nous avons supposé que la garde de la transition $T_{n,q}$ est $g_{n,q} = [a_i \leq x_i \leq b_i]$. Cette transition peut être franchie au plus tôt lorsque l'horloge x_i atteint la valeur a_i.

La condition pour que la transition $T_{n,p}$ soit toujours franchie avant la transition $T_{n,q}$ est exprimée par l'inégalité suivante :

$$a_i - x_i \quad > \quad b_j - x_j \tag{4.7}$$

Cette inégalité représente la condition pou que le temps minimal de séjour du système dans le sommet L_n avant de franchir la transition $T_{n,q}$ soit plus grand que le temps maximal de séjour dans L_n avant de franchir $T_{n,p}$.

Par la suite, on doit vérifier si les valuations des horloges dans le sommet L_n satisfont cette contrainte.

Déterminer l'espace D_n des horloges actives désiré dans L_n : L'ensemble des valuations des horloges dans le sommet L_n qui vérifient la contrainte 4.7 est décrit par l'espace D_n des horloges actives désiré dans ce sommet. Le calcul de cet espace nécessite la connaissance de l'espace des horloges actives dans le sommet L_n.

L'espace des horloges actives dans le sommet L_n, noté $Suc_t(E_n^a)$, est calculé par la méthode donnée lors de la présentation de l'étape *traitement aval*, à la page 100.

L'espace D_n des horloges actives désiré dans le sommet L_n mémorise l'ensemble des valuations des horloges appartenant à l'espace $Suc_t(E_n^a)$ qui satisfont la condition 4.7.

$$D_n \quad = \quad Suc_t(E_n^a) \wedge [a_i - x_i > b_j - x_j] \tag{4.8}$$

Ainsi, on cherche à réduire l'espace atteignable dans le sommet L_n aux valuations des horloges qui vérifient la condition 4.7.

Par la suite, on calcule l'ensemble des valuations des horloges à l'entrée dans le sommet L_n qui ne permettent pas le franchissement de $T_{n,q}$. Cet ensemble est décrit par l'espace des horloges désiré à l'entrée dans L_n.

Calculer l'espace des horloges désiré à l'entrée dans L_n : L'espace E_n^d des horloges désiré à l'entrée dans le sommet L_n est calculé par la relation suivante :

$$E_n^d = E_n \wedge D_n \tag{4.9}$$

L'espace D_n a été calculé d'une manière différente que dans le cas précédent de l'étape traitement aval. Nous montrons que dans ce cas aussi, l'espace E_n^d caractérise toutes les valuations des horloges à l'entrée dans L_n telles que le sommet interdit L_q ne soit pas atteignable.

Propriété 4.2.3. *L'espace E_n^d des horloges désiré à l'entrée dans L_n mémorise toutes les valuations des horloges à l'entrée dans L_n qui garantissent que $T_{n,p}$ est toujours franchie avant $T_{n,q}$.*

Preuve: Soit $v \in E_n^d$ une valuation des horloges à l'entrée dans L_n.

$$\left.\begin{array}{l} v \in E_n^d \\ E_n^d = E_n \wedge D_n \end{array}\right\} \Rightarrow v \in D_n$$

$$\left.\begin{array}{l} v \in D_n \\ D_n = Suc_t(E_n^a) \wedge [a_i - x_i > b_j - x_j] \end{array}\right\} \Rightarrow v \vDash [a_i - x_i > b_j - x_j]$$

La condition 4.7 peut être réécrite de la manière suivante :

$$x_j - x_i > b_j - a_i$$

Nous rappelons que dans un sommet de l'automate temporisé toutes les horloges ont la même vitesse de variation. Ceci fait que la différence $x_j - x_i$ reste constante pendant le séjour du système dans L_n. Par conséquent, tous les successeurs continus de la valuation v vérifient la condition 4.7. On peut conclure que lorsqu'on atteint le sommet L_n avec une valuation $v \in E_n^d$, on quitte toujours le sommet L_n en franchissant la transition $T_{n,p}$ avant que $T_{n,q}$ puisse être franchie.

Considérons maintenant qu'on atteint le sommet L_n avec une valuation des horloges $v' \in E_n$ telle que $v' \notin E_n^d$.

$$\left.\begin{array}{l} v' \in E_n \\ v' \notin E_n^d \\ E_n^d = E_n \wedge D_n \end{array}\right\} \Rightarrow v' \notin D_n$$

Cependant, v' est une valuation des horloges dans le sommet L_n, donc $v \in Suc_t(E_n^a)$.

$$\left.\begin{array}{l} v' \in D_n \\ D_n = Suc_t(E_n^a) \wedge [a_i - x_i > b_j - x_j] \end{array}\right\} \Rightarrow v' \nvDash [a_i - x_i > b_j - x_j]$$

Lorsque la différence $x_j - x_i$ est constante pendant le séjour du système dans L_n, il n'y a aucun successeur temporel de la valuation v' qui vérifie la condition 4.7. Par conséquent, lorsqu'on atteint le sommet L_n avec une valuation $v \notin E_n^d$, le système peut franchir la transition $T_{n,q}$ vers le sommet interdit L_n.

■

Puisqu'on n'a pas eu le droit de modifier la garde de la transition $T_{n,p}$, le sommet interdit L_q reste toujours atteignable. Ainsi, il y a forcément des valuations des horloges dans l'espace E_n qui permettent l'évolution vers ce sommet. Par conséquent, dans le cas où la transition $T_{n,p}$ est incontrôlable, la relation $\mathbf{E_n^d = E_n}$ n'est jamais vérifiée. Il reste donc deux cas possibles :

1. $\mathbf{E_n^d} = \phi$: Alors il n'y a aucune valuation des horloges à l'entrée dans L_n telle que la contrainte 4.7 soit vérifiée pendant le séjour du système dans le sommet L_n. Dans ce cas, il n'y a aucune solution pour forcer le franchissement de la transition $T_{n,p}$ avant celui de $T_{n,q}$. Lorsque $T_{n,p}$ est la seule transition de sortie de L_n vers un sommet non interdit, alors L_n devient lui-même un sommet interdit.
2. $\mathbf{E_n^d} \neq \phi$: Dans ce cas il y a des valuations des horloges dans L_n qui satisfont la contrainte 4.7 mais il y a aussi des valuations qui ne la vérifient pas. Par conséquent, il faut remonter les branches de l'automate et calculer des nouvelles gardes pour les transitions contrôlables telles que toutes les valuations des horloges à l'entrée dans le sommet L_n appartiennent à l'espace E_n^d. Ce traitement est effectué pendant la deuxième étape de la synthèse de la commande, appelée ***traitement amont***. Ainsi, on mémorise dans la pile P le sommet L_n avec l'espace E_n^d des horloges désiré à son entrée.

Nous illustrons ce cas de l'étape traitement aval à travers un exemple.

Exemple 4.3. Considérons la partie de l'automate temporisé illustrée dans la figure 4.6.

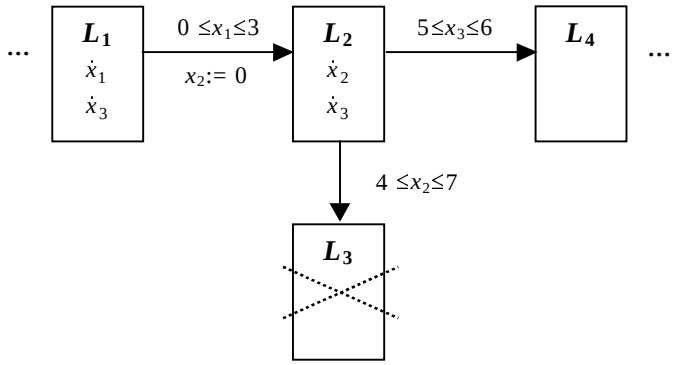


FIG. 4.6 – Partie d'un automate temporisé

Soit L_3 un sommet interdit. Ce sommet est atteint depuis le sommet L_2 par le franchissement de la transition $T_{2,3}$. La garde de cette transition est $g_{2,3} = [4 \leq x_2 \leq 7]$. Cependant, le sommet L_2 a une transition de sortie vers le sommet non-interdit L_4. Nous supposons que cette transition est incontrôlable. Ainsi, la garde de cette transition ne peut pas être modifiée par la synthèse de la commande. Selon la remarque 4.1, la garde de cette transition porte sur la valeur d'une seule horloge.

Soit $g_{2,4} = [5 \leq x_3 \leq 6]$ la garde de $T_{2,4}$.

D'un coté, la transition $T_{2,4}$ doit être franchie au plus tard lorsque l'horloge x_3 atteint la valeur 6. D'un autre coté, la transition $T_{2,3}$ peut être franchie au plus tôt lorsque l'horloge x_2 atteint la valeur 4. Ainsi, la contrainte qu'on doit mettre sur la valeur des horloges pour que la transition $T_{2,4}$ soit toujours franchie avant $T_{2,3}$ est :

$$4 - x_2 > 6 - x_3$$

Par la suite on détermine l'ensemble des valuations des horloges dans le sommet L_2 qui vérifient cette contrainte. D'abord on calcule l'espace des horloges actives dans ce sommet. Cet espace mémorise l'ensemble de toutes les valuations d'horloges dans le sommet L_2. Pour effectuer ce calcul, on a besoin de connaître l'invariant du sommet L_2 et l'espace des horloges actives à l'entrée dans ce sommet.

Supposons que l'invariant du sommet L_2 est :

$$I(L_2) = [0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6].$$

Le sommet L_2 a une seule transition d'entrée. Supposons que l'espace d'entrée dans ce sommet par le franchissement de la transition $T_{1,2}$ est décrit par l'expression suivante :

$$E_{1,2} = [x_1 = x_3 \wedge 0 \leq x_1 \leq 3 \wedge x_2 = 0].$$

L'espace des horloges à l'entrée le sommet L_2 est :

$$E_2 = E_{1,2} = [x_1 = x_3 \wedge 0 \leq x_1 \leq 3 \wedge x_2 = 0]$$

L'espace des horloges actives à l'entrée dans L_2 est obtenu par la projection orthogonale de l'espace E_2 sur la direction des horloges actives dans ce sommet.

$$\begin{aligned} E_2^a &= Pr_{x_2,x_3}(E_2) \\ &= [x_2 = 0 \wedge 0 \leq x_3 \leq 3] \end{aligned}$$

L'espace des horloges actives dans le sommet L_2 est le successeur continu de l'espace E_2^a.

$$\begin{aligned} Suc_t(E_2^a) &= \exists t \in \mathbb{R}^+ . E_2^a[x_2 - t, x_3 - t].I(L_2) \\ &= \exists t \in \mathbb{R}^+ . [0 \leq x_3 - t \leq 3 \wedge x_2 - t = 0] \wedge [0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \\ &= [0 \leq x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \end{aligned}$$

L'ensemble des valuations des horloges dans le sommet L_2 qui vérifient la contrainte $[4 - x_2 > 6 - x_3]$ est mémorisé dans l'espace D_2 des horloges actives désiré dans L_2.

$$\begin{aligned} D_2 &= Suc_t(E_2^a) \wedge [4 - x_2 > 6 - x_3] \\ &= [0 \leq x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \wedge [x_3 - x_2 > 6 - 4] \\ &= [2 < x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 6 \wedge 0 \leq x_3 \leq 5] \\ &\neq \phi \end{aligned}$$

Enfin, on calcule l'espace E_2^d des horloges désiré à l'entrée dans L_2.

$$\begin{aligned} E_2^d &= E_2 \wedge D_2 \\ &= [x_1 = x_3 \wedge 0 \leq x_1 \leq 3 \wedge x_2 = 0] \wedge \\ &\quad [2 < x_3 - x_2 \leq 3 \wedge 0 \leq x_2 \leq 7 \wedge 0 \leq x_3 \leq 6] \\ &= [x_1 = x_3 \wedge 2 < x_1 \leq 3 \wedge x_2 = 0] \end{aligned}$$

Comme prévu, $E_2^d \neq E_2$, donc on peut atteindre ce sommet avec des valuations qui n'appartiennent pas à l'espace D_2.

On mémorise dans une pile P le sommet L_2, les horloges actives dans ce sommet et l'espace E_2^d des horloges à son entrée.

■

Jusqu'à maintenant nous avons considéré que le sommet L_n a une seule transition de sortie vers un sommet non-interdit. Cependant, en général, il peut exister plusieurs transitions de sortie de ce sommet qu'on peut franchir pour ne pas évoluer vers le sommet interdit L_q. Nous détaillons ce cas par la suite.

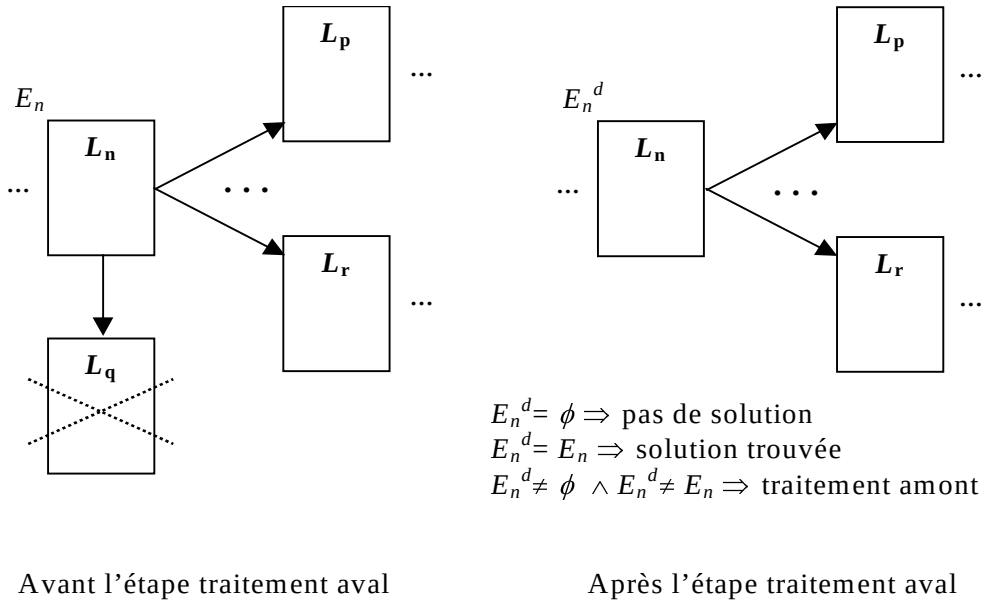


FIG. 4.7 – Principe de général de l'étape traitement aval

Cas général de l'étape traitement aval Supposons que le sommet L_n a plusieurs transitions de sortie $T_{n,l}$, où $l = p..r$ qu'on peut franchir pour quitter ce sommet afin de ne pas évoluer vers le sommet interdit L_q par le franchissement de la transition $T_{n,q}$. Cette situation est illustrée dans la figure 4.7.

Dans ce cas, pour chaque transition $T_{n,l}$ de sortie du L_n vers un sommet non-interdit, on calcule l'espace $(E_n^d)_{n,l}$ des horloges désiré à l'entrée dans L_n. Cet espace mémorise toutes les valuations des horloges à l'entrée dans L_n qui garantissent que la transition $T_{n,l}$ est toujours franchie avant $T_{n,q}$.

L'espace E_n^d des horloges désiré à l'entrée dans L_n est défini par l'union des espaces $(E_n^d)_{n,l}$ calculés pour chaque transition $T_{n,l}$.

$$E_n^d \ = \ \vee_l (E_n^d)_{n,l} \text{ où } l = p..r \tag{4.10}$$

Propriété 4.2.4. *L'espace E_n^d définit l'ensemble de toutes les valuations des horloges à l'entrée dans L_n qui garantissent que le sommet interdit L_q n'est pas atteignable depuis L_n.*

Preuve: Soit $v \in E_n^d$ une valuation des horloges à l'entrée dans L_n.

$$\left.\begin{array}{l} v \in E_n^d \\ E_n^d = \vee_l (E_n)_{n,l},\ l = p..r \end{array}\right\} \Rightarrow \exists l \text{ t.q. } v \in (E_n^d)_{n,l}$$

Par conséquent, selon les propriétés 4.2.2 et 4.2.3, la valuation des horloges v garantit que le système quitte le sommet L_n par le franchissement de $T_{n,l}$ avant que le franchissement de la transition $T_{n,q}$ soit possible.

Par conséquent, lorsqu'on atteint le sommet L_n avec une valuation des horloges $v \in E_n^d$, alors la transition $T_{n,q}$ vers le sommet interdit L_q ne peut plus être franchie.

Supposons maintenant qu'on atteint le sommet L_n avec une valuation des horloges $v' \in E_n$ tel que $v' \notin E_n^d$.

$$\left.\begin{array}{l} v' \in E_n^d \\ E_n^d = \vee_l (E_n)_{n,l},\ l = p..r \end{array}\right\} \Rightarrow \forall l = p..r.\ v' \notin (E_n^d)_{n,l}$$

Selon les propriétés 4.2.2 et 4.2.3, il n'y a aucune transition de sortie $T_{n,l}$ qui soit toujours franchie avant $T_{n,q}$.

Lorsqu'on atteint le sommet L_n avec une valuation $v' \notin E_n^d$, on ne peut pas empêcher le système de franchir la transition $T_{n,q}$ vers le sommet interdit L_q.

■

On peut distinguer trois cas, selon la valeur de l'espace E_n^d :

1. $\mathbf{E_n^d} = \phi$: Dans ce cas il n'y a aucune solution pour empêcher le système de franchir la transition $T_{n,q}$ et donc d'atteindre le sommet interdit L_q depuis le sommet L_n. Le sommet L_n devient lui-même un sommet interdit.
2. $\mathbf{E_n^d} = \mathbf{E_n}$: Il n'est pas possible d'atteindre le sommet L_n avec des valuations d'horloges telles que le franchissement de $T_{n,q}$ soit possible. Dans ce cas on peut conclure qu'on a trouvé l'ensemble des lois de commande telles que le sommet interdit L_q ne soit jamais atteignable depuis L_n.
3. $\mathbf{E_n^d} \neq \phi$ et $\mathbf{E_n^d} \neq \mathbf{E_n}$: Dans ce cas il faut remonter les branches de l'automate et calculer les nouvelles gardes des transitions contrôlables telles que toutes les valuations des horloges à l'entrée dans L_n appartiennent à l'espace E_n^d. Ce traitement sera effectué pendant l'étape suivante de la synthèse de la commande, appelée ***traitement amont***. On mémorise dans la pile P le sommet L_n, les horloges actives dans ce sommet et l'espace des horloges désiré à son entrée E_n^d.

   De même, on mémorise le sommet L_n dans l'ensemble Q des sommets à partir desquels il faut actualiser l'automate.

Remarque 4.2. Si toutes les transitions de sortie du sommet L_n sont incontrôlables, alors le deuxième cas ne peut pas se produire.

L'étape suivante, appelée **traitement amont**, consiste à calculer les nouvelles gardes des transitions contrôlables en amont du sommet L_n tel que l'espace des horloges désiré à l'entrée dans ce sommet soit inclus dans l'espace E_n^d.

Deuxième étape : traitement amont

Nous expliquons le principe de cette étape en nous appuyant sur la partie d'automate temporisé illustrée dans la figure 4.8.

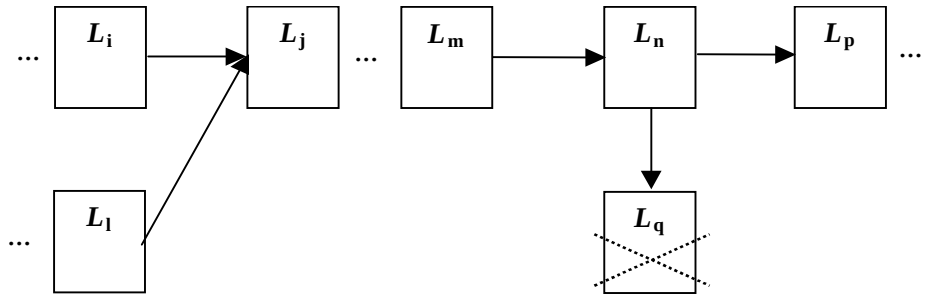


FIG. 4.8 – Partie d'un automate temporisé

Considérons un sommet interdit L_q qui est atteint depuis un sommet L_n par le franchissement de la transition $T_{n,q}$.

Soit E_n^d l'espace des horloges désiré à l'entrée dans le sommet L_n. Cet espace a été calculé pendant l'étape *traitement aval*.

L'objectif de l'étape *traitement amont* est de calculer des nouvelles gardes pour les transitions contrôlables en amont du sommet L_n telles que toutes les valuations des horloges à l'entrée dans ce sommet appartiennent à l'espace E_n^d.

Cet objectif est atteint en appliquant d'une manière itérative les opérations suivantes :

- on remonte une transition de l'automate temporisé. Supposons qu'il s'agit de la transition $T_{i,j}$.
- si $T_{i,j}$ est contrôlable on calcule une nouvelle garde pour cette transition. L'objectif de cette opération est d'obtenir un espace $E_{i,j}$ des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ tel que $E_{i,j} \subseteq E_j^d$.
- on calcule l'espace E_i^d des horloges désiré à l'entrée dans le sommet source de $T_{i,j}$.

On arrête de remonter une branche de l'automate temporisé lorsqu'on atteint un sommet L_i tel que $E_i^d = E_i$.

Par la suite nous détaillons le principe de l'étape *traitement amont*.

Supposons que le sommet analysé soit L_j. L'espace des horloges désiré à son entrée est E_j^d. Supposons que ce sommet a plusieurs transitions d'entrée. Il est possible qu'il ne soit pas nécessaire de remonter toutes ces transitions. Ainsi, dans un premier temps on doit déterminer les transitions qu'il faut remonter.

Soit $T_{i,j}$ une transition d'entrée dans le sommet L_j. L'espace des horloges à l'entrée dans ce sommet par le franchissement de la transition $T_{i,j}$ est $E_{i,j}$.

Si $E_{i,j} \subseteq E_j^d$, alors toutes les valuations des horloges avec lesquelles on peut atteindre le sommet L_j par le franchissement de $T_{i,j}$ appartiennent à l'espace E_j^d. Dans ce cas, ce n'est pas nécessaire de remonter cette transition.

Par contre, si $E_{i,j} \not\subset E_j^d$, alors il y a des valuations des horloges à l'entrée dans L_j par le franchissement de $T_{i,j}$ qui peuvent engendrer une évolution non-désirée. Dans ce cas, il faut remonter cette transition et, si elle est contrôlable, modifier sa garde telle que le nouvel espace des horloges obtenu à l'entrée dans L_j soit inclus dans l'espace E_j^d.

Considérons le cas particulier où L_j est le sommet initial de l'automate temporisé, c'est à dire $L_j = L_0$, et $E_{0,0} \not\subseteq E_0^d$. Lorsqu'il n'est pas possible de remonter la transition $T_{0,0}$, on peut conclure qu'il n'y a aucune solution pour garantir que le sommet interdit L_q n'est pas atteignable depuis le sommet L_n.

Dans le cas général, pour chaque transition $T_{i,j}$ qu'on doit remonter, on effectue les opérations suivantes :

- on calcule l'espace $E_{i,j}^d$ des horloges désiré à l'entrée dans L_n par le franchissement de $T_{i,j}$.
- si la transition $T_{i,j}$ est contrôlable, alors on calcule sa nouvelle garde $g_{i,j}^d$ telle que les valuations possibles pour les horloges à l'entrée dans L_j appartiennent à l'espace $E_{i,j}^d$.
- on calcule l'espace E_i^d des horloges désiré à l'entrée dans L_i, le sommet source de $T_{i,j}$.

Par la suite nous détaillons ces opérations. Supposons qu'il faille remonter la transition $T_{i,j}$. Cette transition peut être contrôlable ou incontrôlable selon la nature de l'événement modélisé. Dans un premier cas nous détaillons le cas où la transition $T_{i,j}$ est contrôlable.

Transition $T_{i,j}$ contrôlable Nous expliquons le déroulement d'une itération de l'étape traitement amont dans le cas où $T_{i,j}$ est contrôlable en considérant la partie d'automate temporisé présentée dans la figure 4.9.

Initialement, l'espace des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ est $E_{i,j} \neq E_{i,j}^d$. La transition $T_{i,j}$ est contrôlable et sa garde $g_{i,j}$ peut être modifiée

lors de la synthèse de la commande. Enfin, l'espace des horloges à l'entrée dans le sommet L_i est E_j. Cette situation est illustrée dans la figure 4.9.a.

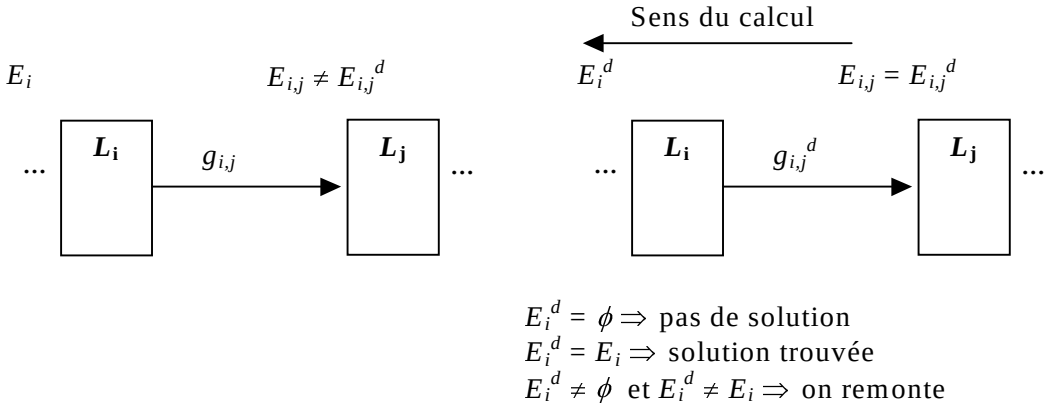
Les objectifs de cette itération de l'étape traitement amont sont :

- déterminer une nouvelle garde $g^d_{i,j}$ de $T_{i,j}$, la moins contraignante, telle que toutes les valuations des horloges à l'entrée dans L_j par le franchissement de $T_{i,j}$ appartiennent à l'espace $E^d_{i,j}$;
- calculer l'espace des horloges désiré à l'entrée dans le sommet L_i.

Pour atteindre ces objectifs nous effectuons les opérations suivantes :

- on détermine la nouvelle garde $g^d_{i,j}$ de la transition $T_{i,j}$;
- on calcule l'espace D_i des horloges actives désiré dans le sommet L_i ;
- on calcule l'espace E^d_i des horloges désiré à l'entrée dans le sommet L_i.

A la fin de cette itération, l'espace des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ est $E_{i,j} = E^d_{i,j}$. La nouvelle garde de $T_{i,j}$ est $g^d_{i,j}$ et l'espace des horloges désiré à l'entrée dans le sommet L_i est E^d_i. Cette situation est illustrée dans la figure 4.9.b.



a. Au début de l'itération du traitement amont

b. A la fin de l'itération du traitement amont

FIG. 4.9 – Partie d'un automate temporisé

Par la suite nous détaillons les opérations effectuées pendant le déroulement d'une itération de l'étape traitement amont dans le cas où la transition à remonter est contrôlable.

Calculer la nouvelle garde $g^d_{i,j}$: Le calcul de la nouvelle garde de la transition $T_{i,j}$ nécessite la connaissance de l'espace des horloges actives dans le sommet L_i. Cet espace, noté $Suc_t(E^a_i)$, est calculé par la méthode donnée lors de la présentation de l'étape *traitement aval*, à la page 100.

Soit $S_{i,j}$ l'ensemble des valuations des horloges dans le sommet L_i qui vérifient la garde $g_{i,j}$. Cet espace est défini par la relation suivante :

$$S_{i,j} = Suc_t(E^a_i) \wedge g_{i,j} \tag{4.11}$$

L'espace des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ est le successeur discret de l'espace $S_{i,j}$ par le franchissement de cette transition.

$$\begin{aligned} E_{i,j} &= Suc_{i,j}(S_{i,j}) \\ &= Suc_{i,j}(Suc_t(E^a_i) \wedge g_{i,j}) \end{aligned} \tag{4.12}$$

Cet espace est obtenu à partir de $S_{i,j}$ par la mise à zéro des horloges spécifiées par l'affectation $A_{i,j}$ associée à $T_{i,j}$. Soit v une valuation des horloges. Nous rappelons que la notation $v[A_{i,j}]$ désigne la valuation des horloges obtenue en mettant à zéro les horloges spécifiées par l'affectation $A_{i,j}$. Cette notation a été présentée dans le chapitre 2, section 2.2.1. Ainsi, l'équation 4.12 peut être réécrite sous la forme suivante :

$$\begin{aligned} E_{i,j} &= Suc_{i,j}(Suc_t(E_i^a) \wedge g_{i,j}) \\ &= (Suc_t(E_i^a) \wedge g_{i,j})[A_{i,j}] \end{aligned} \tag{4.13}$$

Notre objectif est de déterminer la nouvelle garde $g_{i,j}^d$ telle que $E_{i,j} = E_{i,j}^d$.

$$\begin{aligned} E_{i,j}^d &= Suc_{i,j}(Suc_t(E_i^a) \wedge g_{i,j}^d) \\ &= (Suc_t(E_i^a) \wedge g_{i,j}^d)[A_{i,j}] \end{aligned} \tag{4.14}$$

Considérons l'espace $(E_{i,j}^d)'$ obtenu par la projection de l'espace $E_{i,j}^d$ sur les directions des horloges qui sont actives dans L_i mais qui ne sont pas mises à zéro par l'affectation $A_{i,j}$:

$$(E_{i,j}^d)' = Pr_{horloges\ actives\ non-mises\ à\ zéro}(E_{i,j}^d) \tag{4.15}$$

L'espace $E_{i,j}^d$ est obtenu à partir de l'espace $(E_{i,j}^d)'$ en mettant à zéro les horloges spécifiées par l'affectation $A_{i,j}$. Par conséquent, la relation suivante est vraie :

$$E_{i,j}^d = (E_{i,j}^d)'[A_{i,j}] \tag{4.16}$$

Alors, à partir des équations 4.13 et 4.16 on obtient la relation suivante :

$$\begin{aligned} E_{i,j} \wedge E_{i,j}^d &= (Suc_t(E_i^a) \wedge g_{i,j})[A_{i,j}] \wedge E_{i,j}^d \\ &= (Suc_t(E_i^a) \wedge g_{i,j})[A_{i,j}] \wedge (E_{i,j}^d)'[A_{i,j}] \\ &= (Suc_t(E_i^a) \wedge g_{i,j} \wedge (E_{i,j}^d)')[A_{i,j}] \end{aligned} \tag{4.17}$$

De plus, lorsque $E_{i,j}^d \subseteq E_{i,j}$, alors $E_{i,j} \wedge E_{i,j}^d = E_{i,j}^d$. Ainsi, l'équation 4.17 devient :

$$E_{i,j}^d = (Suc_t(E_i^a) \wedge g_{i,j} \wedge (E_{i,j}^d)')[A_{i,j}] \tag{4.18}$$

Enfin, en comparant les équations 4.14 et 4.18 on détermine l'expression de la nouvelle garde de la transition $T_{i,j}$:

$$g_{i,j}^d = g_{i,j} \wedge (E_{i,j}^d)' \tag{4.19}$$

Celle-ci est la garde la moins contraignante de la transition $T_{i,j}$ qui garantit que l'ensemble des valuations des horloges à l'entrée de L_j est l'espace $E_{i,j}^d$.

L'opération suivante à effectuer est le calcul de l'espace E_i^d des horloges désiré à l'entrée dans le sommet L_i.

Calculer l'espace des horloges désiré à l'entrée dans L_i : Ce calcul est exactement le même que celui effectué lors de la présentation de l'étape *traitement aval*, le cas où la transition $T_{n,p}$ est contrôlable (page 100).

Propriété 4.2.5. *L'espace des horloges E_i^d désiré à l'entrée dans L_i définit l'ensemble de toutes les valuations des horloges à l'entrée dans ce sommet qui garantissent qu'on atteint le sommet L_j par le franchissement de $T_{i,j}$ avec un espace des horloges $E_{i,j} \subseteq E_{i,j}^d$.*

Preuve: Soit $v \in E_i$ une valuation des horloges à l'entrée dans L_i telle que $v \notin E_i^d$.

$$\left.\begin{array}{l} v \in E_i \\ v \notin E_i^d \\ E_i^d = E_i \wedge D_i \end{array}\right\} \Rightarrow v \notin D_i$$

Cependant, v est une valuation des horloges dans le sommet L_i. Alors, $v \in Suc_t(E_i^a)$.

$$\left.\begin{array}{l} v \notin D_i \\ v \in Suc_t(E_i^a) \\ D_i = Suc_t(E_i^a) \wedge Pre_t(S_{i,j}^d) \end{array}\right\} \Rightarrow v \notin Pre_t(S_{i,j}^d)$$

Par conséquent, lorsqu'on atteint le sommet L_i avec une valuation $v \notin E_i^d$, l'espace $S_{i,j}^d$ ne peut pas être atteint pendant l'évolution du système dans ce sommet. Donc, la nouvelle garde $g_{i,j}^d = g_{i,j} \wedge (E_{i,j}^d)'$ de la transition $T_{i,j}$ n'est pas satisfaite pendant le séjour du système dans ce sommet. Dans ce cas on ne peut pas atteindre le sommet L_j avec une valuation des horloges incluse dans $E_{i,j}^d$.

Considérons maintenant qu'on atteint le sommet L_i avec une valuation des horloges $v' \in E_i^d$.

$$\left.\begin{array}{l} v' \in E_i^d \\ E_i^d = E_i \wedge D_i \end{array}\right\} \Rightarrow v' \in D_i$$

$$\left.\begin{array}{l} v' \in D_i \\ D_i = Suc_t(E_i^a) \wedge Pre_t(S_{i,j}^d) \end{array}\right\} \Rightarrow v' \in Pre_t(S_{i,j}^d)$$

Par conséquent, lorsqu'on atteint le sommet L_i avec une valuation des horloges $v' \in E_i^d$, l'évolution du système dans ce sommet permet d'atteindre l'espace $S_{i,j}^d$. Ainsi, la nouvelle garde $g_{i,j}^d = g_{i,j} \wedge (E_{i,j}^d)'$ de la transition $T_{i,j}$ est satisfaite et on atteint le sommet L_j avec une valuation des horloges qui appartient à l'espace $E_{i,j}^d$.

■

On peut distinguer trois cas, selon la valeur de l'espace E_i^d :

1. $\mathbf{E_i^d} = \phi$: Dans ce cas il n'y a aucune valuation des horloges possible à l'entrée dans le sommet L_i telle que $E_{i,j} = E_{i,j}^d$. Ainsi, on peut conclure qu'il n'y a aucune solution pour garantir que le sommet interdit L_q n'est plus atteignable depuis le sommet L_n. Le sommet L_n devient lui-même un sommet interdit.
2. $\mathbf{E_i^d} = \mathbf{E_i}$ Toutes les valuations possibles à l'entrée dans le sommet L_i appartiennent nécessairement à l'espace des horloges désiré à l'entrée dans ce sommet. Ainsi, il n'est pas nécessaire de remonter au-delà du sommet L_i.
3. $\mathbf{E_i^d} \neq \phi$ et $\mathbf{E_i^d} \neq \mathbf{E_i}$: Il faut continuer à remonter. On mémorise dans la pile P le sommet L_i avec l'espace des horloges désiré à son entrée. Il fera l'objet d'une prochaine itération.

   De même, on actualise l'ensemble Q des sommets à partir desquels il faut actualiser l'automate. Ainsi, on enlève de cet ensemble le sommet destination de $T_{i,j}$, i.e. L_j et on rajoute le sommet L_i, qui est son sommet source.

L'association d'une nouvelle garde à la transition $T_{i,j}$ peut déterminer une nouvelle valeur maximale pour la valeur de certaines horloges dans le sommet L_i. Ainsi, il faut actualiser l'invariant du L_i pour prendre en compte cette modification.

Le nouvel invariant de ce sommet est calculé à travers les opérations suivantes :

- d'abord, on détermine la valeur maximale que chaque horloge x_k, active dans L_i peut prendre pendant le séjour du système dans ce sommet :

$$(x_k)_{max} = max\{Pr_{x_k}(D_i)\} \tag{4.20}$$

- ensuite, on calcule le nouvel invariant du L_i :

$$I(L_i) \quad := \quad I(L_i) \wedge_k [0 \leq x_k \leq (x_k)_{max}] \; \forall x_k \text{ horloge active dans } L_i \tag{4.21}$$

Nous illustrons le déroulement de cette étape à travers un exemple.

Exemple 4.4. Considérons la partie d'automate temporisé présentée dans la figure 4.10.

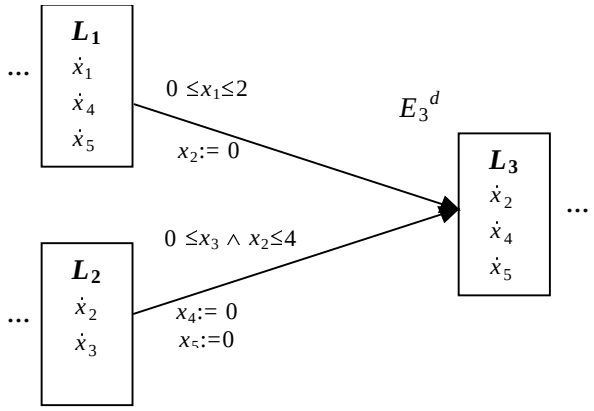


FIG. 4.10 – Partie d'un automate temporisé

Supposons que pendant l'itération courante de l'étape traitement amont on analyse le sommet L_3 avec l'espace des horloges désiré à son entrée $E_3^d \neq E_3$. Ce sommet a deux transitions d'entrée : $T_{1,3}$ et $T_{2,3}$. Par conséquent, l'espace des horloges à son entrée est :

$$E_3 = E_{1,3} \vee E_{2,3}$$

L'espace des horloges E_3^d désiré à l'entrée dans le sommet L_3 a été calculé lors d'une itération précédente.

$$\begin{aligned} E_3^d &= E_3 \vee D_3 \\ &= [E_{1,3} \wedge D_3] \vee [E_{2,3} \wedge D_3] \end{aligned}$$

Supposons que E_3^d est décrit par l'expression :

$$\begin{aligned} E_3^d = \; & [0 \leq x_1 - x_4 \leq 2 \wedge x_4 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge \\ & 0 \leq x_4 \leq 1 \wedge x_2 = 0 \wedge x_2 - x_4 < 3] \vee \\ & [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_4 = 0 \wedge x_5 = 0] \end{aligned}$$

Ce sommet a deux transitions d'entrée : $T_{1,3}$ et $T_{2,3}$. Par conséquent, il faut déterminer la(les) transition(s) qu'on doit remonter.

D'abord on analyse s'il faut remonter la transition $T_{1,3}$. L'espace des horloges à l'entrée dans L_3 par le franchissement de $T_{1,3}$ est

$$\begin{aligned} E_{1,3} = \; & [0 \leq x_1 - x_4 \leq 2 \wedge x_4 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge \\ & \wedge 0 \leq x_4 \leq 1 \wedge x_2 = 0] \end{aligned}$$

Pour déterminer s'il faut remonter la transition $T_{1,3}$, on vérifie si $E_{1,3} \subseteq E_3^d$.

$$\begin{aligned} E_{1,3} \wedge E_3^d &= [0 \leq x_1 - x_4 \leq 2 \wedge x_4 = x_5 \wedge 0 \leq x_1 \leq 2 \wedge \\ &\quad \wedge 0 \leq x_4 \leq 1 \wedge x_2 = 0] \\ &= E_{1,3} \end{aligned}$$

Ainsi, il n'est pas nécessaire de remonter cette transition.

Ensuite on analyse la nécessité de remonter la transition $T_{2,3}$. L'espace des horloges à l'entrée dans L_3 par le franchissement de $T_{2,3}$ est :

$$E_{2,3} = [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 4 \wedge x_4 = 0 \wedge x_5 = 0]$$

On vérifie si $E_{1,3} \subseteq E_3^d$.

$$\begin{aligned} E_{2,3} \wedge E_3^d &= [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_4 = 0 \wedge x_5 = 0] \\ &\neq E_{2,3} \end{aligned}$$

Par conséquent, $E_{2,3} \not\subseteq E_{2,3}^d$, donc il faut remonter la transition $T_{2,3}$.

Considérons que la transition $T_{2,3}$ est contrôlable. Dans ce cas cette itération de l'étape traitement amont consiste à faire les opérations suivantes :

- déterminer la nouvelle garde $g_{2,3}^d$ de $T_{2,3}$;
- calculer l'espace E_2^d des horloges désiré à l'entrée dans L_2.

Calculer la nouvelle garde $g_{2,3}^d$: L'espace des horloges désiré à l'entrée dans L_3 par le franchissement de $T_{2,3}$ est :

$$\begin{aligned} E_{2,3}^d &= E_{2,3} \wedge E_3^d \\ &= [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_4 = 0 \wedge x_5 = 0] \end{aligned}$$

La projection de cet espace sur les directions des horloges qui sont actives dans L_2, mais qui ne sont pas mises à zéro par l'affectation $A_{2,3}$ est :

$$\begin{aligned} (E_{2,3}^d)' &= Pr_{x_2,x_3}(E_{2,3}^d) \\ &= [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

La nouvelle garde de la transition $T_{2,3}$ est calculée par la relation suivante :

$$\begin{aligned} g_{2,3}^d &= g_{2,3} \wedge (E_{2,3}^d)' \\ &= [0 \leq x_3 \wedge x_2 < 4] \wedge \\ &\quad [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \\ &= [0 \leq x_3 \leq 5 \wedge x_2 < 3] \end{aligned}$$

Par la suite nous calculons l'espace des horloges actives désiré dans le sommet L_2.

Calculer l'espace des horloges actives désiré D_2 : Le calcul de cet espace nécessite la connaissance de l'espace des horloges actives dans le sommet L_2. Nous avons déjà illustré le principe de calcul de cet espace. Ainsi, nous donnons directement la relation qui le décrit :

$$Suc_t(E_2^a) = [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_3 < 6 \wedge 0 \leq x_2 < 4]$$

Cet espace décrit l'ensemble des valuations qu'on peut atteindre dans le sommet L_2.
L'espace de valuations des horloges dans le sommet L_2 qui vérifient la garde $g^d_{2,3}$ est :

$$\begin{aligned} S^d_{2,3} &= Suc_t(E^a_2) \wedge g^d_{2,3} \\ &= [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_3 < 5 \wedge 0 \leq x_2 < 3] \end{aligned}$$

L'ensemble des valuations des horloges qu'on souhaite atteindre dans L_2 est décrit par le prédécesseur temporel de l'espace $S^d_{2,3}$.

$$Pre_t(S^d_{2,3}) = [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_3 < 5 \wedge 0 \leq x_2 < 3]$$

Enfin, l'espace des horloges actives désiré dans le sommet L_2 est :

$$\begin{aligned} D_2 &= Pre_t(S^d_{2,3}) \wedge Suc_t(E^a_2) \\ &= [0 \leq x_3 - x_2 \leq 2 \wedge 0 \leq x_3 < 5 \wedge 0 \leq x_2 < 3] \end{aligned}$$

Calculer l'espace des horloges désiré à l'entrée dans L_2 : Considérons que l'espace des horloges à l'entrée dans le sommet L_2 est :

$$E_2 = [x_1 = x_3 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0]$$

L'espace des horloges désiré à l'entrée dans L_2 est :

$$\begin{aligned} E^d_2 &= E_2 \wedge D_2 \\ &= [x_1 = x_3 \wedge 0 \leq x_1 \leq 2 \wedge x_2 = 0] \end{aligned}$$

Puisque $E^d_2 = E_2$ on arrête de remonter cette branche.
La modification de la garde associée à la transition $T_{2,3}$ peut déterminer une nouvelle valeur maximale pour la valeur des horloges x_2 et x_3 dans le sommet source de cette transition.
Supposons que l'invariant du sommet L_2 est :

$$I(L_2) = [0 \leq x_2 < 4 \wedge 0 \leq x_3 < 6] \tag{4.22}$$

Ainsi, il faut actualiser l'invariant du sommet L_2 pour prendre en compte les changements engendrées par l'association d'une nouvelle garde à $T_{2,3}$.
D'abord on calcule la valeur maximale que chacune des horloges x_2 et x_3 peut prendre dans L_2.

$$\begin{aligned} (x_2)_{max} &= max\{Pr_{x_2}(D_2)\} \\ &= max\{0 \leq x_2 < 3\} \end{aligned}$$

$$\begin{aligned} (x_3)_{max} &= max\{Pr_{x_3}(D_2)\} \\ &= max\{0 \leq x_2 < 3\} \end{aligned}$$

Ensuite on calcule le nouvel invariant du L_2 :

$$\begin{aligned} I(L_2)^d &= I(L_2) \wedge [0 \leq x_2 < 3] \wedge [0 \leq x_3 < 5] \\ &= [0 \leq x_2 < 3 \wedge 0 \leq x_3 < 5] \end{aligned}$$

Le déroulement de l'étape traitement amont continue avec l'analyse de l'élément suivant mémorisé dans la pile P.

■

Par la suite nous présentons le cas où la transition à remonter $T_{i,j}$ est incontrôlable.

Transition $T_{i,j}$ incontrôlable Nous détaillons le déroulement d'une itération de l'étape traitement amont dans le cas où $T_{i,j}$ est incontrôlable en nous appuyant sur la partie de l'automate temporisé illustrée dans la figure 4.11.

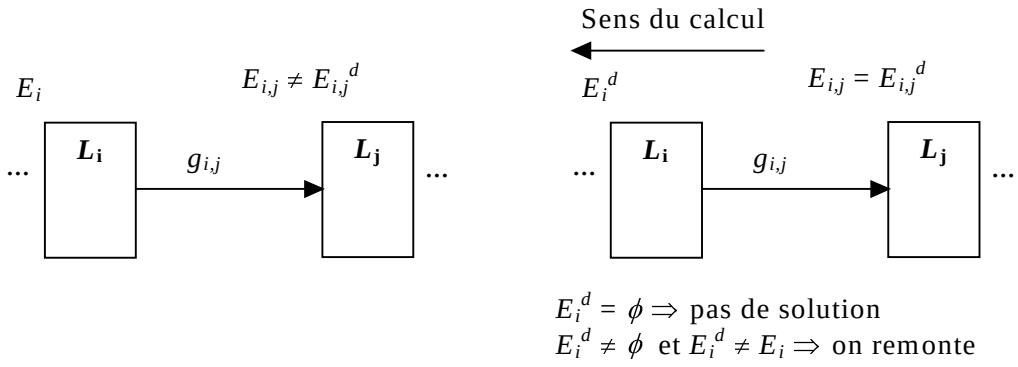
Initialement, l'espace des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ est $E_{i,j} \neq E^d_{i,j}$. La transition $T_{i,j}$ est incontrôlable, donc sa garde $g_{i,j}$ ne peut pas être modifiée lors de la synthèse de la commande. De plus, selon la remarque 4.1, la garde de cette transition porte sur la valeur d'une seule horloge. Soit $g_{i,j} = [a_k \leq x_k \leq b_k]$ la garde de cette transition. Enfin, l'espace des horloges à l'entrée dans le sommet L_i est E_i. Cette situation est illustrée dans la figure 4.11.a.

Notre objectif est de calculer l'espace des horloges E^d_i désiré à l'entrée dans L_i tel que l'espace des horloges à l'entrée dans L_j par le franchissement de $T_{i,j}$ soit $E_{i,j} = E^d_{i,j}$.

Cet objectif est atteint à travers les opérations suivantes :

- on détermine l'espace D_i des horloges actives désiré dans le sommet L_i.
- on calcule ensuite l'espace des horloges désiré à l'entrée dans le sommet L_i.

A la fin de cette itération l'espace des horloges à l'entrée dans le sommet L_j par le franchissement de $T_{i,j}$ est $E_{i,j} = E^d_{i,j}$. L'espace des horloges désiré à l'entrée dans le sommet L_i est E^d_i. Cette situation est illustrée dans la figure 4.11.b.



a. Au début de l'itération du traitement amont

b. A la fin de l'itération du traitement amont

FIG. 4.11 – Partie d'un automate temporisé

Par la suite nous détaillons les opérations associées à une itération de l'étape traitement amont dans le cas où la transition à remonter est incontrôlable.

Calculer l'espace des horloges actives désiré D_i : Le calcul de cet espace nécessite la connaissance de l'espace des horloges actives dans le sommet L_i. Cet espace, noté $Suc_t(E^a_i)$, est calculé par la méthode donnée lors de la présentation de l'étape *traitement aval*, à la page 100.

Ensuite on calcule l'ensemble des valuations des horloges dans le sommet L_i qui vérifient la garde $g_{i,j}$ et à partir desquelles on atteint l'espace $E^d_{i,j}$ par le franchissement de $T_{i,j}$. Cet espace, noté $S^d_{i,j}$, est le prédécesseur discret de l'espace $E^d_{i,j}$ par le franchissement de $T_{i,j}$.

$$S^d_{i,j} \quad = \quad Pre_{i,j}(E^d_{i,j}) \tag{4.23}$$

L'ensemble des valuations des horloges désiré dans L_2 est formé par deux types de valuations :

- valuations qui appartiennent à l'espace $S^d_{i,j}$;
- valuations à partir desquelles on atteint l'espace $S^d_{i,j}$ en laissant le temps s'écouler.

Cet ensemble est le prédécesseur temporel de l'espace $S^d_{i,j}$, noté $Pre_t(S^d_{i,j})$. Il mémorise les valuations d'horloges qu'on souhaite atteindre pendant le séjour du système dans le sommet L_i.

L'espace des horloges actives désiré dans le sommet L_i est défini par l'intersection de l'espace des valuations des horloges existant dans L_i, i.e $Suc_t(E^a_i)$, et l'espace des valuations qu'on souhaite atteindre dans ce sommet, i.e. $Pre_t(S^d_{i,j})$:

$$D_i \quad = \quad Pre_t(S^d_{i,j}) \wedge Suc_t(E^a_i) \tag{4.24}$$

Calculer l'espace des horloges désiré à l'entrée dans L_i : L'espace des horloges désiré à l'entrée dans L_i est :

$$E^d_i = E_i \wedge D_i \tag{4.25}$$

Propriété 4.2.6. *L'espace E^d_i des horloges désiré à l'entrée dans L_i mémorise toutes les valuations des horloges à l'entrée dans ce sommet qui garantissent qu'on atteint le sommet L_j par le franchissement de $T_{i,j}$ avec un espace des horloges $E_{i,j} \subseteq E^d_{i,j}$.*

Preuve: Soit $v \in E^d_i$ une valuation des horloges à l'entrée dans L_i.

$$\left.\begin{array}{l} v \in E^d_i \\ E^d_i = E_i \wedge D_i \end{array}\right\} \Rightarrow v \in D_i$$

$$\left.\begin{array}{l} v \in D_i \\ D_i = Suc_t(E^a_i) \wedge Pre_t(S^d_{i,j}) \end{array}\right\} \Rightarrow v \in Pre_t(S^d_{i,j})$$

$$\left.\begin{array}{l} v \in Pre_t(S^d_{i,j}) \\ S^d_{i,j} = Pre_{i,j}(E^d_{i,j}) \end{array}\right\} \Rightarrow v \in Pre_t(Pre_{i,j}(E^d_{i,j}))$$

Par conséquent, lorsqu'on atteint le sommet L_i avec une valuation $v \in E^d_i$, alors on atteint le sommet L_j avec une valuation des horloges appartenant à l'espace $E^d_{i,j}$.

Considérons maintenant qu'on atteint le sommet L_i avec une valuation des horloges $v' \in E_i$ telle que $v' \notin E^d_i$.

$$\left.\begin{array}{l} v' \in E_i \\ v' \notin E^d_i \\ E^d_i = E_i \wedge D_i \end{array}\right\} \Rightarrow v' \notin D_i$$

Cependant, v' est une valuation des horloges dans le sommet L_i, donc $v \in Suc_t(E^a_i)$.

$$\left.\begin{array}{l} v' \in D_i \\ D_i = Suc_t(E^a_i) \wedge Pre_t(S^d_{i,j}) \end{array}\right\} \Rightarrow v' \notin Pre_t(S^d_{i,j})$$

$$\left.\begin{array}{l} v' \notin Pre_t(S^d_{i,j}) \\ S^d_{i,j} = Pre_{i,j}(E^d_{i,j}) \end{array}\right\} \Rightarrow v' \notin Pre_t(Pre_{i,j}(E^d_{i,j}))$$

Lorsqu'on atteint le sommet L_i avec une valuation $v' \notin E^d_i$ alors il n'est pas possible d'entrer dans L_j par le franchissement de $T_{i,j}$ avec des valuations des horloges appartenant à $E^d_{i,j}$.

■

Dans le cas où la transition $T_{i,j}$ est incontrôlable, la relation $\mathbf{E^d_i = E_i}$ n'est jamais vérifiée. Il reste donc deux cas possibles :

1. $\mathbf{E_i^d} = \phi$: Alors, il n'y a aucune valuation des horloges à l'entrée dans L_i qui permette d'atteindre le sommet L_j avec des valuations appartenant à l'espace $E_{i,j}^d$. Par conséquent, il n'y a pas de solution pour garantir que le sommet interdit L_q n'est plus atteignable depuis L_n.
2. $\mathbf{E_i^d} \neq \phi$: Dans ce cas il faut continuer à remonter. On mémorise dans la pile P le sommet L_i, les horloges actives dans ce sommet et l'espace E_i^d des horloges désiré à son entrée. De même, on actualise l'ensemble Q des sommets à partir desquels il faut actualiser l'automate. Ainsi, on enlève de cet ensemble le sommet destination de $T_{i,j}$, i.e. L_j et on rajoute son sommet source, c'est à dire L_i.

Remarque 4.3. La pile P contient l'ensemble des sommets à partir desquels il faut remonter et calculer des nouvelles gardes pour les transitions contrôlables en amont de ces sommets.

L'étape traitement amont se termine si l'une des situations suivantes est rencontrée :
1) la pile est vide, donc il ne reste aucun sommet à partir duquel il faut remonter ;
2) en remontant on atteint le sommet initial L_0 avec un espace des horloges désiré à son entrée E_0^d tel que $E_{0,0} \not\subseteq E_0^d$.

■

Pendant le déroulement des deux premières étapes de la synthèse de la commande, i.e. le traitement aval et le traitement amont, les gardes de certaines transitions de l'automate temporisé ont été modifiées. Ces changements peuvent rendre certains sommets non-atteignables. Ainsi, après avoir calculé des nouvelles gardes pour certaines transitions contrôlables, il faut actualiser l'automate temporisé afin de prendre en compte les effets de ces modifications. D'un coté, cette opération est motivée par le souci de représenter dans le modèle seulement les évolutions réalisables. D'un autre coté, lors de la synthèse de la commande pour un certain sommet interdit il est possible de rendre non-atteignables, indirectement, d'autres sommets interdits. Ce serait complètement inutile d'appliquer la procédure de synthèse de la commande pour un sommet interdit qui de toute façon n'est plus atteignable.

Par conséquent, la troisième étape de la synthèse de la commande consiste à actualiser l'automate temporisé. Cette étape est présentée par la suite.

Troisième étape : Actualiser l'automate temporisé

Soit L_j un sommet mémorisé dans l'ensemble Q lors des étapes traitement aval ou traitement amont. L'actualisation de l'automate à partir de ce sommet consiste à descendre les branches de l'automate d'une manière itérative et à supprimer les éléments qui ne sont plus atteignables. A chaque itération on analyse les transitions de sortie d'un sommet. Le principe de cette opération est illustré dans la figure 4.12.

Supposons que le sommet analysé pendant l'itération courante est L_n. Les opérations effectuées pendant cette itération sont :

- on détermine les transitions qui ne sont plus franchissables depuis ce sommet et on élimine tous les éléments de l'automate temporisé qui sont rendus non-atteignables. Par exemple, si la transition $T_{n,q}$ ne peut plus être franchie, alors on l'élimine ainsi que son sommet destination L_q.
- pour les transitions qui restent franchissables depuis ce sommet, on actualise l'espace des horloges à l'entrée dans leur sommet destination. Par exemple, si la transition $T_{n,p}$ reste franchissable, on actualise l'espace $E_{n,p}$ des horloges à l'entrée dans le sommet L_p.

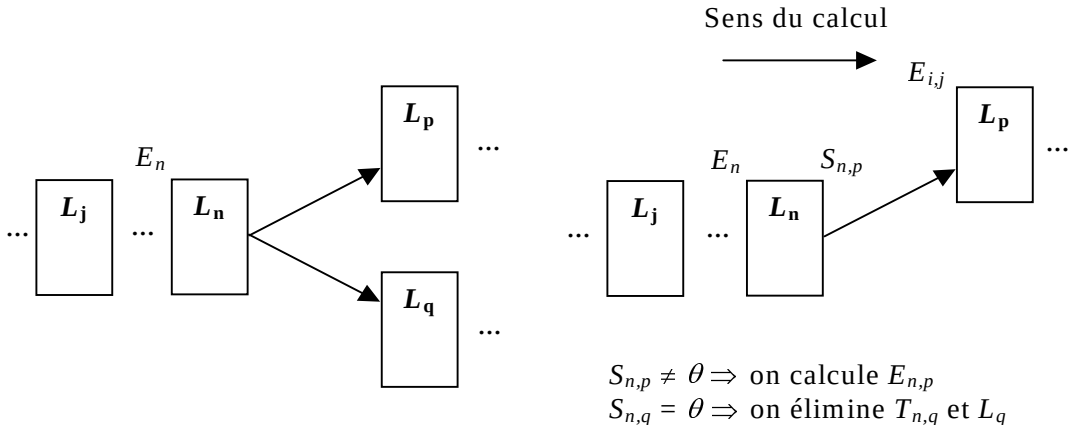


FIG. 4.12 – Partie d'un automate temporisé

Par la suite nous détaillons ces opérations :

Déterminer les transitions qui ne sont plus franchissables Pour déterminer si une transition $T_{n,p}$ est franchissable depuis un sommet L_n on doit vérifier s'il y a des valuations des horloges dans ce sommet qui vérifient la garde $g_{n,p}$ de cette transition.

Dans un premier temps il faut déterminer l'ensemble des valuations qu'on peut atteindre dans le sommet L_n. Nous rappelons que cet ensemble est décrit par l'espace des horloges actives dans le sommet L_n. Cet espace, noté $Suc_t(E_n^a)$ est calculé par la méthode donnée lors de la présentation de l'étape *traitement aval*, à la page 100.

Ensuite, on calcule l'ensemble des valuations des horloges dans le sommet L_n qui vérifient la garde $g_{n,p}$ de la transition $T_{n,p}$. Cet ensemble, noté $S_{n,p}$, est décrit par la relation :

$$S_{n,p} = Suc_t(E_n^a) \wedge g_{n,p} \tag{4.26}$$

Si $\mathbf{S_{n,p}} = \phi$, alors la garde $g_{n,p}$ n'est pas vérifiée par aucune valuation des horloges dans L_n. Dans ce cas, la transition $T_{n,p}$ ne peut plus être franchie depuis ce sommet. Par conséquent, les éléments de l'automate temporisé qui étaient atteints suite au franchissement de cette transition sont rendus non-atteignables. Donc on élimine de la structure de l'automate la transition $T_{n,p}$ ainsi que tous les éléments qui ne sont plus atteignables.

Par contre, si $\mathbf{S_{n,p}} \neq \phi$, alors il y a des valuations des horloges dans L_n telles que la garde $g_{n,p}$ soit vérifiée. Ainsi, la transition $T_{n,p}$ reste franchissable. Dans ce cas il faut calculer l'espace des horloges à l'entrée le sommet L_p par le franchissement de cette transition.

Calculer l'espace des horloges à l'entrée dans L_p par le franchissement de $T_{n,p}$: L'espace des horloges à l'entrée dans le sommet L_p par le franchissement de $T_{n,p}$ pour la visite courante est le successeur discret de l'espace $S_{n,p}$ par le franchissement de cette transition.

$$e_{n,p} = Suc_{n,p}(S_{n,p}) \tag{4.27}$$

L'espace des horloges à l'entrée dans L_p par le franchissement de $T_{n,p}$ pour toutes les visites, noté $E_{n,p}$, est calculé de la même manière que dans l'algorithme de passage du

RdP T-temporel vers les automates temporisés.

Si on visite le sommet L_p pour la première fois pendant le déroulement cette étape de actualisation de l'automate, alors l'espace $E_{n,p}$ est calculé par la relation :

$$E_{n,p} := e_{n,p} \tag{4.28}$$

Par contre, si le sommet L_p a été déjà visité, alors l'espace $E_{n,p}$ est l'union des espaces des horloges à l'entrée dans L_p par le franchissement de $T_{n,p}$ pour toutes les visites :

$$E_{n,p} := E_{n,p} \vee e_{n,p} \tag{4.29}$$

On actualise également l'espace des horloges à l'entrée dans L_p. Cet espace, noté E_p, est défini par l'union des espaces des horloges obtenus à l'entrée dans L_p par le franchissement de chacune de ces transitions d'entrée.

On mémorise dans une pile P le sommet L_p avec l'espace des horloges à son entrée E_n.

L'actualisation de l'automate temporisé suite au changement des gardes des transitions est la dernière étape de la méthode que nous proposons pour la synthèse de la commande.

Par la suite, nous présentons l'algorithme que nous proposons pour la synthèse de la commande. Cet algorithme regroupe les trois étapes que nous venons de présenter.

4.2.2 Algorithme de synthèse de la commande

Avant d'introduire l'algorithme de synthèse de la commande, nous rappelons les notations utilisées :

- $e_{m,n}$: l'espace des horloges à l'entrée dans le sommet L_n par le franchissement de la transition $T_{m,n}$, pour la visite courante ;
- $E_{m,n}$: l'espace des horloges à l'entrée dans le sommet L_n par le franchissement de la transition $T_{m,n}$, pour toutes les visites ;
- $E^d_{m,n}$: l'espace des horloges désiré à l'entrée dans le sommet L_n par le franchissement de $T_{m,n}$;
- E_n : l'espace des horloges à l'entrée dans le sommet L_n. Cet espace mémorise les valeurs que les horloges peuvent prendre lorsque le système atteint le sommet L_n par le franchissement de n'importe quelle transition d'entrée dans ce sommet. Il est défini par l'union des espaces $E_{m,n}$ obtenus à l'entrée dans L_n par le franchissement de chacune de ses transitions d'entrée $T_{m,n}$.
- E^a_n : l'espace des horloges actives à l'entrée dans le sommet L_n ;
- E^d_n : l'espace des horloges désiré à l'entrée dans le sommet L_n ;
- D_n : l'espace des horloges actives désiré dans le sommet L_n ;
- $g^d_{m,n}$ est la nouvelle garde calculée pour la transition $T_{m,n}$;
- $S_{m,n}$: l'espace des horloges actives dans le sommet L_m qui vérifient la garde $g_{m,n}$ de la transition de sortie $T_{m,n}$;
- $S^d_{m,n}$: l'espace des horloges actives dans le sommet L_m qui vérifient la nouvelle garde $g^d_{m,n}$ de la transition de sortie $T_{m,n}$;
- Q : l'ensemble des sommets à partir desquels il faut actualiser l'automate ;
- F : l'ensemble des sommets interdits ;
- P : une pile qui mémorise les visites des sommets non encore analysées. Chaque élément de la pile mémorise les caractéristiques d'une visite d'un sommet :
  - le nom du sommet ;
  - les horloges actives dans ce sommet ;

– l'espace d'horloges à son entrée.

Une partie de ces notations a été utilisée également dans le chapitre 3, lors de la construction de l'automate temporisé qui modélise le comportement d'un RdP T-temporel.

Algorithme de synthèse de la commande Initialement, l'ensemble F mémorise tous les sommets interdits. Les autres ensembles sont vides.

Pas 1 : *Initialisation*

1.1. Si $F = \phi$, il ne reste aucun sommet interdit **Allez au pas 5**.

1.2. Soit $L_q \in F$ un sommet interdit. Si L_q est le sommet initial, alors le problème de synthèse n'a pas de solution. **Allez au pas 5**

1.3. Soit L_n un sommet à partir duquel on atteint le sommet interdit L_q par le franchissement d'une transition $T_{n,q}$.

Si L_n n'a aucune transition de sortie vers un sommet non-interdit, alors L_n devient lui-même un sommet interdit :

- On supprime les transitions de sortie du sommet L_n, ainsi que les éléments de l'automate qui ne sont plus atteignables.
- On actualise l'ensemble des sommets interdits :
  - on ajoute le sommet L_n à l'ensemble F :
$$F := F \cup \{L_n\}$$
  - si parmi les sommets supprimés de l'automate il y a des sommets interdits, alors on les enlève également de l'ensemble F.
- **Allez au pas 1**

SINON On initialise la pile et l'ensemble des sommets à partir desquels il faut actualiser l'automate :

- $Q := \phi$;
- $P := \phi$.

Pas 2 : *Traitement aval*

2.1. Calculer l'espace des horloges à l'entrée dans le sommet L_n
$$E_n = \vee_m E_{m,n}$$
$\forall\ T_{m,n}$ transition d'entrée dans le sommet L_n.

2.2. Calculer l'espace des horloges actives dans le sommet L_n :

- On calcule l'espace des horloges actives à son entrée :
$$E_n^a = Pr_{horloges\ actives}(E_n)$$
- On calcule l'espace des horloges actives dans L_n. Cet espace est le successeur continu de l'espace des horloges actives à son entrée. Il est noté $Suc_t(E_n^a)$.

2.3. Soit $g_{n,q} = [a_i \leq x_i \leq b_i]$ la garde de la transition $T_{n,q}$ qui mène vers le sommet interdit L_q.

Pour chaque transition $T_{n,l}$ qui mène vers un sommet non-interdit, on calcule l'espace des horloges désiré à l'entrée dans L_n. Cet espace, noté $(E_n^d)_{n,l}$, mémorise les valuations des horloges à l'entrée dans L_n telles que la transition $T_{n,l}$ soit toujours franchie avant $T_{n,q}$.

SI $T_{n,l}$ **est contrôlable**, faire :

- Calculer la nouvelle garde $g_{n,l}^d$ de cette transition :
$$g_{n,l}^d = g_{n,l} \wedge [x_i < a_i]$$
- Calculer l'espace des horloges actives désiré dans L_n :
  - calculer l'ensemble des valuations dans L_n qui vérifient la garde $g_{n,l}^d$:
$$S_{n,l}^d = Suc_t(E_n^a) \wedge g_{n,l}^d$$
  - calculer l'espace des horloges actives désiré dans L_n par rapport à $T_{n,l}$:

$$(D_n)_{n,l} = Suc_t(E_n^a) \wedge Pre_t(S_{n,l}^d)$$

- Calculer l'espace des horloges désiré à l'entrée dans L_n par rapport à $T_{n,l}$:

$$(E_n^d)_{n,l} = E_n \wedge (D_n)_{n,l}$$

SINON ($T_{n,l}$ est incontrôlable), faire :

Soit $g_{n,l} = [a_j \leq x_j \leq b_j]$ la garde de la transition $T_{n,l}$.

- Calculer la condition sur la valeur des horloges actives dans L_n :

$$a_i - x_i > b_j - x_j$$

- Calculer l'espace des horloges actives désiré dans L_n par rapport à $T_{n,l}$:

$$(D_n)_{n,l} = Suc_t(E_n^a) \wedge [a_i - x_i > b_j - x_j]$$

- Calculer l'espace des horloges désiré à l'entrée dans L_n par rapport à $T_{n,l}$:

$$(E_n^d)_{n,l} = E_n \wedge (D_n)_{n,l}$$

2.4. Calculer l'espace des horloges désiré à l'entrée dans L_n :

$$E_n^d = \vee_l (E_n^d)_{n,l}$$

Si $\mathbf{E_n^d} = \phi$, alors le sommet L_n devient lui-même un sommet interdit.

- Supprimer de l'automate les transitions de sortie de L_n, ainsi que les éléments qui sont plus atteignables.
- Actualiser l'ensemble des sommets interdits :
  - on ajoute le sommet L_n à l'ensemble F :

$$F := F \cup \{L_n\}$$

  - si parmi les sommets supprimés de l'automate il y a des sommets interdits, alors on les enlève également de l'ensemble F.
- **Allez au pas 1**

Si $\mathbf{E_n^d} = \mathbf{E_n}$, alors on a trouvé la solution pour rendre le sommet L_q non-atteignable depuis L_n. Cependant, il faut actualiser l'automate temporisé :

- On met à jour l'invariant du sommet L_n :

$$I(L_n)^d \quad = \quad I(L_n) \wedge [x_i < a_i]$$

- On met à jour l'ensemble des sommets à partir desquels il faut actualiser l'automate :

$$Q := \{L_n\}$$

- **Allez au pas 4**

Si $\mathbf{E_n^d} \neq \mathbf{E_n}$ et **$\mathbf{E_n^d} \neq \phi$**, alors il faut remonter dans l'automate :

- On met à jour l'invariant du sommet L_n :

$$I(L_n)^d \quad = \quad I(L_n) \wedge [x_i < a_i]$$

- On met à jour l'ensemble des sommets à partir desquels il faut actualiser l'automate :

$$Q := \{L_n\}$$

- On mémorise dans la pile P le sommet L_n, les horloges actives dans ce sommet et l'espace E_n^d :

$$P := \{[L_n; \text{horloges actives}; E_n^d]\}$$

Pas 3 : *Traitement amont*

On analyse le dernier élément introduit dans la pile. Supposons qu'il s'agisse du sommet L_j avec l'espace des horloges à son entrée E_j^d.

3.1. Enlever cet élément de la pile :

$$P := P \setminus \{[L_j; \text{horloges actives}; E_j^d]\}$$

3.2. On détermine les transitions d'entrée dans L_j qu'on doit remonter.

Pour chaque transition $T_{i,j}$ d'entrée dans le sommet L_j on calcule l'espace des horloges $E_{i,j}^d$ à l'entrée dans le sommet L_j par son franchissement.

$$E^d_{i,j} = E_{i,j} \wedge E^d_j$$

SI $E_{i,j} \subseteq E^d_j$, alors il n'est pas nécessaire de remonter la transition $T_{i,j}$.
SINON il faut remonter la transition $T_{i,j}$.

3.3. Pour chaque transition $T_{i,j}$ à remonter, faire :

- Calculer l'espace des horloges à l'entrée dans le sommet source de $T_{i,j}$, i.e. L_i :
$$E_i = \vee_h E_{h,i}$$
$\forall T_{h,i}$ transition d'entrée dans le sommet L_i.
- Calculer l'espace des horloges actives dans le sommet L_i :
  - on calcule l'espace des horloges actives à son entrée :
  $$E^a_i = Pr_{horloges\ actives}(E_i)$$
  - on calcule l'espace des horloges actives dans L_i. Cet espace est le successeur continu de l'espace des horloges actives à son entrée. Il est noté $Suc_t(E^a_i)$.

SI $T_{i,j}$ **est contrôlable**, faire :

- Calculer la nouvelle garde $g^d_{i,j}$:
  - calculer l'espace $(E^d_{i,j})'$ qui est la projection de l'espace $E^d_{i,j}$ sur les dimensions des horloges actives dans L_i qui ne sont pas mises à zéro par l'affectation associée à la transition $T_{i,j}$.
  $$(E^d_{i,j})' = Pr_{horloges\ actives\ non-mises\ à\ zéro}(E^d_{i,j}) \tag{4.30}$$
  - Calculer la nouvelle garde $g^d_{i,j}$:
  $$g^d_{i,j} = g_{i,j} \wedge (E^d_{i,j})'.$$
- Calculer l'espace des horloges désiré dans L_i
  - Calculer l'ensemble des valuations des horloges dans L_i qui vérifient la garde $g^d_{i,j}$:
  $$S^d_{i,j} = Suc_t(E^a_i) \wedge g^d_{i,j}.$$
  - Calculer l'espace des horloges actives désiré dans L_i :
  $$D_i = Suc_t(E^a_i) \wedge Pre_t(S^d_{i,j})$$
- Calculer l'espace des horloges désiré à l'entrée dans le sommet L_i :
$$E^d_i = E_i \wedge D_i$$

Si $\mathbf{E^d_i} = \phi$, alors le sommet L_n devient lui-même un sommet interdit.

- Supprimer de l'automate les transitions de sortie de L_n, ainsi que les éléments qui ne sont plus atteignables.
- Actualiser l'ensemble des sommets interdits :
  - on ajoute le sommet L_n à l'ensemble F :
  $$F := F \cup \{L_n\}$$
  - si parmi les sommets supprimés de l'automate il y a des sommets interdits, on les enlève également de l'ensemble F.
- **Allez au pas 1**

Si $\mathbf{E^d_i} = \mathbf{E_i}$, alors il n'est pas nécessaire de remonter au delà du sommet L_i.

- On met à jour l'invariant du sommet L_i :
  - on détermine la valeur maximale de horloge active x_k dans le sommet L_i :
  $$(x_k)_{max} \quad = \quad max\{Pr_{x_k}(D_i)\}$$
  - le nouvel invariant du L_i est :
  $$I(L_i)^d = I(L_i) \wedge_k [0 \leq x_k \leq (x_k)_{max}]\ \forall x_k \text{ horloge active dans } L_i \tag{4.31}$$
- On met à jour l'ensemble des sommets à partir desquels il faut actualiser l'au-

tomate :

$$\begin{aligned} Q &:= Q \setminus \{L_j\} \\ &:= Q \cup \{L_i\} \end{aligned}$$

Si $\mathbf{E_i^d \neq E_i}$ et $\mathbf{E_i^d \neq \phi}$, alors il faut remonter dans l'automate.

- On met à jour l'invariant du sommet L_i :
  - on détermine la valeur maximale de horloge active x_k dans le sommet L_i :

$$(x_k)_{max} = max\{Pr_{x_k}(D_i)\}$$

  - le nouvel invariant du L_i est :

$$I(L_i)^d = I(L_i) \wedge_k [0 \leq x_k \leq (x_k)_n] \ \forall x_k \text{ horloge active dans } L_i \qquad (4.32)$$

- On met à jour l'ensemble des sommets à partir desquels il faut actualiser l'automate :

$$\begin{aligned} Q &:= Q \setminus \{L_j\} \\ &:= Q \cup \{L_i\} \end{aligned}$$

- On mémorise dans la pile P le sommet L_i, les horloges actives dans ce sommet et l'espace E_i^d :

$$P := P \cup \{[L_i; \text{horloges actives}; E_i^d]\}$$

SINON($T_{i,j}$ est incontrôlable), faire :

- Calculer l'espace des horloges désiré dans L_i :
  - calculer l'ensemble des valuations des horloges dans L_i qui vérifient la garde $g_{i,j}$:

$$S_{i,j} = Suc_t(E_i^a) \wedge g_{i,j}.$$

  - calculer l'espace des horloges actives désiré dans L_i :

$$D_i = Suc_t(E_i^a) \wedge Pre_t(S_{i,j})$$

- Calculer l'espace des horloges désiré à l'entrée dans le sommet L_i :

$$E_i^d = E_i \wedge D_i$$

Si $\mathbf{E_i^d = \phi}$, alors le sommet L_n devient lui-même un sommet interdit.

- Eliminer de l'automate les transitions de sortie de L_n, ainsi que les éléments qui ne sont plus atteignables.
- Actualiser l'ensemble des sommets interdits :
  - on ajoute le sommet L_n à l'ensemble F :

$$F := F \cup \{L_n\}$$

  - si parmi les sommets supprimés de l'automate il y a des sommets interdits, alors on les enlève également de l'ensemble F.
- **Allez au pas 1**

Si $\mathbf{E_i^d \neq E_i}$ et $\mathbf{E_i^d \neq \phi}$, alors il faut remonter dans l'automate.

- On met à jour l'ensemble des sommets à partir desquels il faut actualiser l'automate :

$$\begin{aligned} Q &:= Q \setminus \{L_j\} \\ &:= Q \cup \{L_i\} \end{aligned}$$

- On mémorise dans la pile P le sommet L_i, les horloges actives dans ce sommet et l'espace E_i^d :

$$P := P \cup \{[L_i; \text{horloges actives}; E_i^d]\}$$

3.4 Si $P \neq \phi$, **Allez au pas 3**

Remarque 4.4. A la fin de l'étape traitement amont, la pile P est vide. De plus l'information qu'on mémorise dans chaque itération de la procédure d'actualisation de l'automate a la même structure dans les étapes précédentes. On peut donc utiliser cette même pile pendant l'actualisation de l'automate.

Pas 4 : *Actualiser l'automate*

On actualise l'automate depuis chaque sommet mémorisé dans l'ensemble Q.

4.1. Soit L_n un sommet mémorisé dans l'ensemble Q.

- Supprimer le sommet L_n de l'ensemble Q :
$$Q := Q \setminus \{L_n\}$$
- Calculer l'espace des horloges à l'entrée dans le sommet L_n :
$$E_n = \vee_m E_{m,n}$$
$\forall\ T_{m,n}$ transition d'entrée dans le sommet L_n.
- On mémorise dans la pile P le sommet L_n, les horloges actives dans ce sommet et l'espace E_n.

4.2. On analyse le dernier élément introduit dans la pile P. Supposons qu'il s'agisse du sommet L_k avec l'espace des horloges à son entrée E_k.

- Calculer l'espace des horloges actives dans L_k :
  - on calcule l'espace des horloges actives à son entrée :
  $$E_k^a = Pr_{horloges\ actives}(E_k)$$
  - on calcule l'espace des horloges actives dans L_k. Cet espace est le successeur continu de l'espace des horloges actives à son entrée. Il est noté $Suc_t(E_k^a)$.

4.3. On détermine les transitions de sortie de L_k qui ne sont plus franchissables.
Pour chaque transition de sortie $T_{k,l}$ du sommet L_k, faire :

- Calculer l'ensemble des valuations des horloges dans L_k qui satisfont la garde $g_{k,l}$ de cette transition :
$$S_{k,l} = Suc_t(E_k^a) \wedge g_{k,l}$$
SI $S_{k,l} = \phi$, alors la transition $T_{k,l}$ ne peut pas être franchie depuis L_k :
- On élimine $T_{k,l}$ de l'automate, ainsi que les éléments de l'automate temporisé qui ont été rendus non-atteignables.
- Si parmi les éléments supprimés de l'automate il y a aussi des sommets interdits, alors on les enlève également de l'ensemble F.
SINON $T_{k,l}$ reste franchissable
- On calcule l'espace des horloges à l'entrée dans L_l par le franchissement de $T_{k,l}$ pour cette visite :
$$e_{k,l} = Suc_{k,l}(S_{k,l})$$
- On actualise l'espace des horloges à l'entrée dans L_l par le franchissement de $T_{k,l}$ pour toutes les visites.
  - SI c'est la première visite, alors :
  $$E_{k,l} = e_{k,l}$$
  - SINON
  $$E_{k,l} = E_{k,l} \vee e_{k,l}$$
- Calculer l'espace des horloges E_l à l'entrée dans L_l par le franchissement de toutes ses transitions d'entrée.
$$E_l = \vee_k E_{k,l}$$
$\forall T_{k,l}$ transition d'entrée dans le sommet L_l.
SI L_l a au moins une transition de sortie, alors on mémorise dans la pile P le

sommet L_l, les horloges actives dans ce sommet et l'espace E_l.

4.4. Si $P \neq \phi$, **Allez au pas 4.2.**

4.5. Si $Q \neq \phi$, **Allez au pas 4.1.**

Pas 5 : *Fin*

Remarque 4.5. Le nombre d'itérations de l'algorithme dépend de l'ordre dans laquelle on traite les sommets interdits.

■

Par la suite, nous appliquons cet algorithme pour la synthèse de la commande du poste de collage présenté dans la section 2.1.5.

Exemple 4.5. Considérons à nouveau l'exemple du poste de collage présenté dans la section 2.1.5 Nous rappelons que le fonctionnement de ce système a été modélisé par le RdP T-temporel illustré dans la figure 2.18. Dans le chapitre 3, nous avons construit l'automate temporisé qui modélise le comportement de ce RdP T-temporel. L'automate obtenu est représenté dans la figure 3.10. La synthèse de la commande pour le poste de collage est effectuée à partir de cet automate, en appliquant l'algorithme que nous venons d'introduire. Pour améliorer la qualité de la présentation nous reprenons cet automate temporisé dans la figure 4.13.

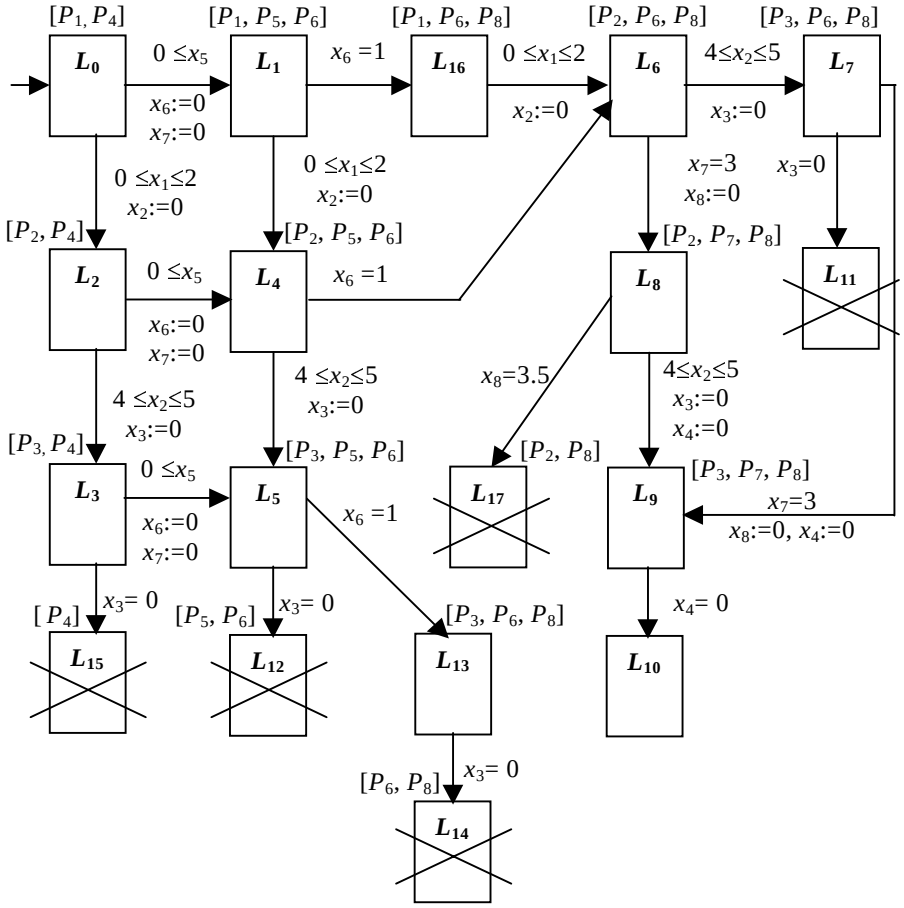


FIG. 4.13 – Automate temporisé du poste de collage

Les événements contrôlables qui peuvent avoir lieu dans ce système sont le démarrage de la tâche de l'opérateur et le démarrage de la tâche du robot. Les autres événements sont incontrôlables.

L'exécution de l'événement démarrage de la tâche de l'opérateur est modélisée au niveau de l'automate temporisé par les transitions $T_{0,1}$ et $T_{2,4}$. L'exécution de l'événement démarrage de la tâche de robot est modélisée dans l'automate temporisé par les transitions $T_{0,2}$ et $T_{16,6}$. Ainsi, les transitions contrôlables de l'automate temporisé associé au poste de collage sont : $T_{0,1}$, $T_{2,4}$, $T_{0,2}$ et $T_{16,6}$. Les autres transitions sont incontrôlables.

Nous rappelons également que les sommets interdits de cet automate sont : L_{11}, L_{12}, L_{14}, L_{15} et L_{17}.

L'objectif de la synthèse de la commande concernant le poste de collage est de calculer les nouvelles gardes des transitions $T_{0,1}$, $T_{2,4}$, $T_{0,2}$ et $T_{16,6}$ telles que les sommets interdits ne soient plus jamais atteints.

Nous présentons l'exécution de l'algorithme pour la synthèse de la commande du poste de collage dans l'annexe A. En appliquent cet algorithme, on obtient l'automate temporisé illustré dans la figure 4.14.

Cet automate représente toutes les lois de commande qui garantissent un collage réussi. Lorsque le système est dans le sommet L_0, ni l'opérateur, ni le robot n'ont pas encore démarré leur tâches. Considérons, par exemple, que le robot démarre sa tâche avant l'opérateur. Cette situation correspond au franchissement de la transition $T_{0,2}$. L'affectation associée à cette transition met à zéro l'horloge x_2. Cette horloge compte le temps écoulé depuis l'instant de démarrage de la tâche du robot. Le sommet atteint par le franchissement de T_2 est L_2. Après la synthèse, ce sommet a une seule transition de sortie dont la garde est $g_{2,4} = [0 \leq x_2 - x_5 \leq 2 \wedge 0 \wedge x_2 < 1 \wedge 0 \leq x_5 < 3]$. Donc, pour obtenir un collage réussi, l'opérateur doit commencer sa tâche avant que 1 u.t. ne se soit écoulée depuis le démarrage de la tâche du robot ($x_2 < 1$) et 3 u.t. ne se soient écoulées depuis le démarrage du processus de collage ($x_5 < 3$). Nous rappelons que le robot commence sa tâche au plus tard 2 u.t. depuis le démarrage du processus de collage. Donc il y a au plus 2 u.t. de décalage entre l'instant de démarrage de l'horloge x_5 et celui de l'horloge x_2. Cette information est illustrée par l'expression $0 \leq x_5 - x_2 \leq 2$.

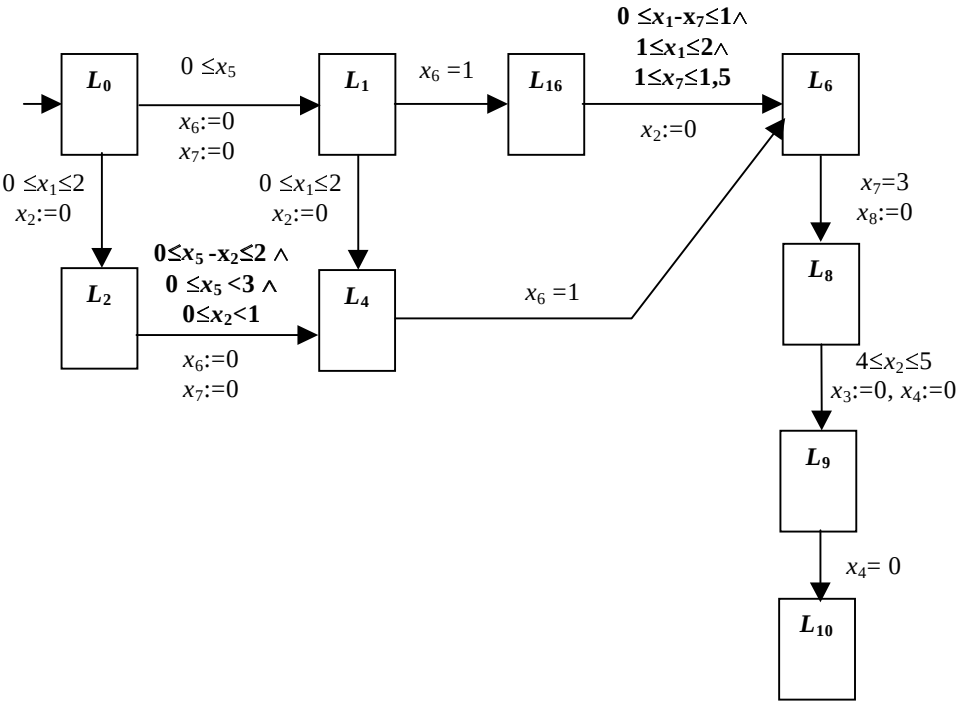


FIG. 4.14 – Automate temporisé du système de commande

Afin de donner une présentation explicite du modèle de commande obtenu, nous remplaçons les gardes de certaines transitions par les événements correspondants dans le procédé. Supposons que les capteurs placés dans le procédé nous permettent d'observer les événements suivants :

1) fin tâche robot, noté fr ;
2) fin tâche opérateur, noté fo ;
3) collage.

L'automate obtenu est illustré dans la figure 4.15. Dans cette figure, les gardes ont été simplifiés en gardant la contrainte la plus forte.

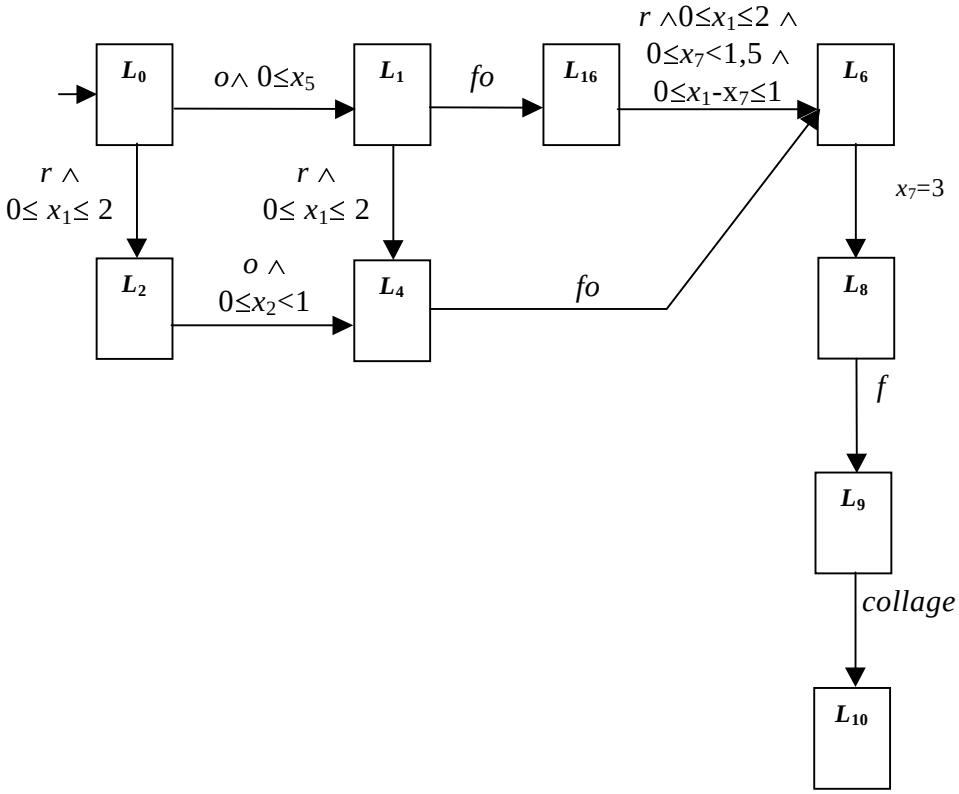


FIG. 4.15 – Modèle de commande du poste de collage

Les durées associées aux événements incontrôlables (provenant des capteurs) qui ont été utilisées, sont des mesures de modélisation. Elles ont été nécessaires pour la synthèse mais n'apparaissent pas forcément dans le modèle de commande.

On peut ainsi noter que seulement les durées fournies par les horloges x_1, x_2 et x_7 sont nécessaires pour la commande du poste de collage :

- l'horloge x_1 est utilisée pour décider de la date de lancement de la tâche du robot. Cet événement doit avoir lieu au plus tard 2 u.t. après le démarrage du processus de collage ;
- l'horloge x_2 compte le temps écoulé depuis le démarrage de la tâche du robot ;
- l'horloge x_7 est utilisée pour décider si la colle est devenue prête pour le collage.

Lorsque le robot démarre le premier (chemin $L_0 \rightarrow L_2 \rightarrow L_4$), alors l'opérateur doit démarrer sa tâche au plus tard 1 u.t. après.

Par contre, si l'opérateur commence le premier (chemin $L_0 \rightarrow L_1 \rightarrow L_4$), alors le robot doit démarrer au plus tard lorsque 1,5 u.t. se soit écoulées depuis le démarrage de la tâche de l'opérateur.

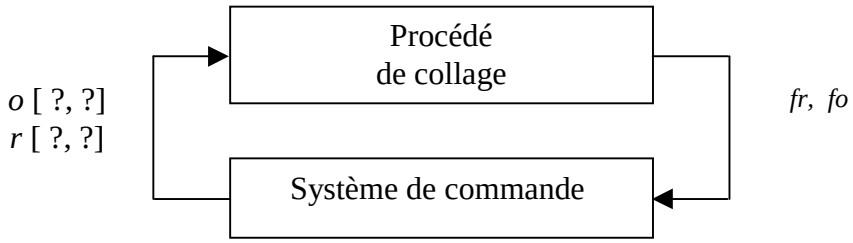


FIG. 4.16 – La synthèse de la commande pour le poste de collage

Le schéma de commande du processus de collage est illustré dans la figure 4.16.

Les dates de démarrage des tâches du robot et de l'opérateur sont calculées d'une façon dynamique par le système de commande, selon l'état du procédé.

■

4.2.3 Analyse de l'algorithme

Proposition 4.2.7. *L'algorithme proposé pour la synthèse de la commande se termine dans un nombre fini de pas pour des automates temporisés avec un nombre fini de sommets et donc les gardes des transitions sont spécifiées par des contraintes qui comparent la valeur des horloges à des nombres entiers.*

Preuve: Nous montrons d'abord que chaque étape de l'algorithme que nous proposons pour la synthèse de la commande se termine en un nombre fini de pas. Ensuite nous montrons également que l'algorithme de synthèse de la commande hérite de cette propriété.

Dans un premier temps nous justifions que le nombre des espaces des horloges distinctes possibles à l'entrée dans chaque sommet est fini.

Les gardes de l'automate temporisé considéré dans notre approche de synthèse de la commande sont spécifiées par des contraintes qui comparent la valeur des horloges avec des nombres rationnels. On peut se ramener aux nombres entiers en multipliant chaque nombre rationnel par le plus grand multiple commun des dénominateurs. On peut considérer, sans perdre la généralité de la preuve qu'au niveau des gardes les horloges sont comparées toujours avec des nombres entiers. Ainsi, les contraintes qui décrivent l'espace d'horloges à l'entrée dans un sommet sont toujours spécifiées par des nombres entiers. Dans ce cas on montre dans [Yov98] que l'espace de la valeur des horloges peut être partagé en un nombre fini de régions élémentaires disjointes. De plus le nombre des régions élémentaires est fini.

Par conséquent, chaque espace d'horloges à l'entrée dans un sommet peut être décrit par l'union d'un nombre fini de régions élémentaires disjointes. Ainsi, le nombre des espaces d'horloges distincts à l'entrée dans un sommet est fini.

Par la suite nous montrons que chaque étape de l'algorithme de synthèse de la commande se termine dans un nombre fini de pas.

L'étape traitement aval se termine dans un nombre fini de pas : Cette étape consiste à calculer l'espace des horloges désiré à l'entrée dans un sommet d'un nombre de fois égal au nombre de transitions de sortie de ce sommet. La procédure de calcul de cet espace est finie. Lorsque le nombre des sommets de l'automate est fini, alors le nombre des transitions de sortie d'un sommet est également fini. Par conséquent, on peut conclure que l'étape traitement aval se termine dans un nombre fini de pas.

L'étape traitement amont se termine dans un nombre fini de pas : Cette étape consiste à remonter certaines branches de l'automate et à réduire l'espace des horloges à l'entrée dans chaque sommet rencontré. Lorsque le nombre des espaces des horloges possibles à l'entrée dans un sommet est fini, alors pendant cette étape on peut visiter chaque sommet un nombre fini de fois. De plus, le nombre des sommets de l'automate est fini. Par conséquent, l'étape traitement amont se termine en un nombre fini de pas.

L'étape d'actualisation de l'automate se termine en un nombre fini de pas : Cette étape consiste à descendre les branches de l'automate, éliminer les éléments qui ne sont plus atteignables et actualiser l'espace des horloges à l'entrée dans chaque sommet. lorsque le nombre des sommets de l'automate est fini, cette étape termine également dans un nombre fini de pas.

L'algorithme proposé pour la synthèse de la commande consiste à appliquer, pour chaque sommet interdit, une fois l'étape traitement aval, une fois l'étape traitement amont et une fois l'étape d'actualisation de l'automate. Lorsque le nombre des sommets interdits est fini, cet algorithme se termine également en un nombre fini de pas.

■

Proposition 4.2.8. *L'algorithme que nous avons proposé pour la synthèse de la commande fournit toutes les lois de commande qui garantissent que les sommets interdits ne sont pas atteints pendant l'évolution de l'automate.*

Preuve: Cette propriété est une conséquence des propriétés 4.2.4, 4.2.5, 4.2.6.

■

4.3 Conclusion

Ce chapitre nous a permis d'introduire la méthode que nous proposons pour la synthèse de la commande d'un SEDT.

Dans un premier temps nous avons proposé une nouvelle classification des événements qui peuvent intervenir dans le fonctionnement d'un SEDT. Nous considérons que ces événements peuvent être classifiés en deux catégories : contrôlables et incontrôlables. Nous considérons qu'un événement est contrôlable si on peut fixer sa date d'occurrence dans un intervalle donné. Par contre, un événement est incontrôlable si on ne peut pas agir sur sa date d'occurrence. Cette classification nous est apparue plus intuitive et plus appropriée pour la commande des SEDT.

Ensuite, nous avons introduit l'algorithme que nous proposons pour la synthèse de la commande. Cet algorithme est basé sur le calcul des nouvelles conditions de franchissement, i.e. gardes, des transitions contrôlables telles que les sommets interdits ne soient plus jamais atteints.

L'automate temporisé obtenu par cet algorithme caractérise toutes les lois de commande qui garantissent que le SEDT commandé respecte les spécifications imposées. De plus, cet automate est minimal dans le sens où il représente seulement les évolutions possibles à partir de chaque sommet.

Dans le chapitre suivant nous présentons les conclusions ainsi que les perspectives du travail de recherche présenté dans ce rapport.

Conclusions et perspectives

L'objectif du travail présenté dans ce mémoire est de développer une méthodologie de synthèse de la commande des SEDT qui permet de répondre à ce problème dans le cas général.

Pour atteindre cet objectif, nous avons proposé une approche qui associe la capacité de modélisation de l'outil RdP T-temporel à la capacité d'analyse des automates temporisés. Cette approche est composée de trois étapes :

1) Dans un premier temps, nous modélisons le SEDT à commander par un modèle RdP T-temporel. Cet outil a retenu notre attention grâce à deux propriétés. D'un coté, il permet de modéliser la plupart des contraintes temporelles intervenant dans le fonctionnement d'un SEDT. D'un autre coté, son évolution est déterminée par l'occurrence d'un seul type d'événements : le franchissement des transitions. La conséquence de cette propriété est une très bonne lisibilité des modèles fournis par cet outil. Le modèle RdP T-temporel que nous considérons dans notre travail n'est pas sauf. Ainsi, il est très approprié pour la représentation des systèmes de production.

2) Ensuite, on modélise le comportement du RdP T-temporel associé au SEDT considéré par un automate temporisé. Cette représentation fournit un modèle clair pour son comportement admissible.

Une première contribution du travail présenté dans ce mémoire est le développement d'un algorithme pour construire d'une manière systématique l'automate temporisé qui modélise le comportement d'un RdP T-temporel. L'idée de cet algorithme est de construire l'ensemble des états atteignables. Ainsi, à chaque marquage du RdP T-temporel considéré on associe un sommet de l'automate temporisé. Pour chaque sommet, on calcule l'espace atteignable. Cet espace définit l'ensemble des valeurs que les horloges actives peuvent prendre pendant le séjour du système dans le sommet. L'algorithme développé converge pour des RdP T-temporels bornés, avec des intervalles de franchissement associés aux transitions spécifiés par des nombres rationnels.

L'automate temporisé construit avec cet algorithme a les caractéristiques suivantes :

- structure minimale dans le sens où seulement les évolutions possibles dans le SEDT sont représentées ;
- absence des incohérences temporelles ;
- simplicité des gardes associées aux transitions, ce qui simplifie beaucoup la synthèse de la commande.

Les comportements non désirés d'un SEDT sont représentés par des sommets interdits de l'automate temporisé.

3) La méthode proposée pour la synthèse de la commande consiste à analyser un par un chaque sommet interdit et à calculer des nouvelles gardes pour les transitions contrôlables

telles que ce sommet ne soit plus atteignable. Ce calcul commence par une analyse en aval (procédure *traitement aval*). Si cette étape ne suffit pas pour donner la solution, on continue avec une analyse en arrière (procédure *traitement amont*). Nous insistons sur le fait que notre méthode cherche à calculer des nouvelles gardes pour toutes les transitions contrôlables en amont d'un sommet interdit. Ainsi, nous proposons une solution globale pour le problème de synthèse de la commande. De plus, cette méthode traite le cas général où les horloges sont couplées.

Notre démarche permet de déterminer toutes les trajectoires admissibles pour le fonctionnement du système. Ensuite, il revient au technologue de choisir la trajectoire qui convient au mieux aux objectifs de la production.

Les perspectives que nous envisageons pour notre travail de recherche sont les suivantes :

1. Une perspective immédiate pour la continuation du travail de recherche présenté dans ce mémoire est la validation de l'algorithme proposé pour la commande d'un système de production sans stock et sans attente. L'application envisagée est l'atelier de traitement des surfaces présenté dans [Che99].

Un travail est déjà en cours de déroulement pour atteindre cet objectif. Dans un premier temps nous considérons le cas particulier d'un atelier avec une seule ligne de traitement. Un tel système est composé d'une station de chargement, d'un ensemble de cuves de traitement, d'une station de déchargement et d'un système de transport représenté par un robot. La durée d'immersion d'une pièce dans une cuve de traitement est comprise entre une valeur minimale et une valeur maximale. Le traitement d'une pièce dans une cuve est considéré terminé si elle y est reste au moins pendant une période de temps égale à la durée minimale d'immersion. Par contre, si la durée maximale d'immersion est dépassée, alors la pièce devient un rebut.

L'objectif de la synthèse de la commande pour un atelier de traitement de surfaces est de déterminer l'ensemble des ordonnancements pour le système de transport (le robot) qui garantissent qu'il n'y a pas de rebut.

Une première étape de ce travail a été effectuée pendant le stage de DEA de R. Abbou [Abb00]. Ce travail a consisté dans la modélisation d'une ligne de traitement de surfaces par un modèle RdP T-temporel.

Ensuite, nous avons construit l'automate temporisé associée au RdP T-temporel modélisant une ligne de traitement de surfaces avec 3 cuves de traitement et deux pièces. L'automate temporisé obtenu a 180 sommets dont 88 sont des sommets interdits.

2. Face à un tel volume de calcul, il nous est apparu absolument nécessaire d'implémenter les algorithmes proposés avant de continuer le travail de validation. Ceci constitue une première perspective.

3. De même il est important d'étudier la complexité des algorithmes proposés.

4. Une autre perspective pour la continuation du travail présenté dans ce mémoire est d'étudier la possibilité d'étendre l'approche proposée pour la synthèse de la commande des systèmes hybrides. Une première généralisation consisterait à ne plus considérer des horloges dans les sommets de l'automate, mais des dynamiques continues de type $\dot{x}=$ constante. La classe des systèmes que l'on peut étudier est alors élargie. Il s'agira alors de voir si les résultats présentés dans ce mémoire sur le plan de l'analyse et de la synthèse peuvent être généralisés. Ceci fait l'objet d'une nouvelle thèse de recherche au laboratoire.

Bibliographie

[Abb00] R. Abbou. Modélisation et analyse des systèmes sans stock et sans attente. exemple d'application : Ligne de traitement de surfaces. Mémoire de DEA, Laboratoire d'Automatique de Grenoble, 2000.

[ACH+95] R. Alur, C. Courcoubetis, N. Halbawacs, T.A. Henzinger, P.H. Ho, X. Nicolin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138 :3–34, 1995.

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126 :183–235, 1994.

[AD98a] H. Alla and R. David. Continuous and hybrid systems. *International Journal of Circuits and Systems*, 8(1) :159–188, 1998.

[AD98b] H. Alla and R. David. A modeling and analysis tool for discrete event systems : Continuous petri net. *Performance Evaluation*, 33 :175–199, 1998.

[AGP+99] K. Altisen, G. Gößler, A. Puneli, J. Sifakis, S. Tripakis, and S. Yovine. A framework for scheduler synthesis. *Proceedings of the 1999 IEEE Real-Time Systems Symposium, Phoenix AZ USA*, December 1999.

[All98] M. Allam. *Sur l'analyse quantitative des RdP hybrides : une approche basée sur les automates hybrides.* PhD thesis, Laboratoire d'Automatique de Grenoble - Institut National Polytechnique de Grenoble, 1998.

[Alu99] R. Alur. Timed automata. *11th International Conference on Computer-Aided Verification, LNCS*, 1633 :8–22, 1999.

[AMPS98] E. Asarin, O. Maler, A. Puneli, and J. Sifakis. Control synthesis for timed automata. *Proc. IFAC Symposium on System Structure and Control*, pages 469–474, 1998.

[BD91] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time petri nets. *IEEE Transactions on Software Engineering*, 17(3) :259–273, March 1991.

[BW94] B. Brandin and W.M. Wonham. Supervisory control of timed discrete event systems. *IEEE Transactions on Automatic Control*, 39(2) :329–341, 1994.

[CA97] S. Caramihai and H. Alla. Supervisory synthesis oprimisation for timed discrete event system. *IFAC Conference on Management and Control, Campinas*, 1997.

[Car97] S. Caramihai. *Superviseur intelligent pour le pilotage en temps réel des systèmes flexibles de fabrication.* PhD thesis, Université Politehnica de Bucarest - Roumanie, 1997.

[Cha96] F. Charbonnier. *Commande supervisée des systèmes à événements discrets.* PhD thesis, Laboratoire d'Automatique de Grenoble-Institut National Polytechnique de Grenoble, 1996.

[Che99] F. Chetouane. *Sur la robustesse dans les systèmes de production. Application à la conduite d'un atelier de traitement de surfaces.* PhD thesis, Laboratoire d'Automatique de Grenoble-Institut National Polytechnique de Grenoble, 1999.

[COTY96] C.Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool kronos. *In Hybrid Systems III, Verification and Control, LNCS*, 1066, 1996.

[DA92] R. David and H. Alla. *Du Grafcet aux réseaux de Petri.* Hermes, Paris, 1992.

[Dav91] R. David. Modeling of dynamic systems by petri nets. *European Control Conference*, July 1991.

[DH94] R. David and H.Alla. Petri nets for modeling of dynamic systems - a survey. *Automatica*, 30(2) :175–202, 1994.

[Gal97] L. Gallon. *Le modèle réseaux de Petri temporisés stochastiques : extensions et applications.* PhD thesis, Université Paul Sabatier, Toulouse, 1997.

[Gou99] A. Gouin. *Contribution à la commande supervisée des systèmes à événements discrets temporisés : synthèse de superviseur dans le cadre du modèle automate.* PhD thesis, LISA - Université d'Angers, 1999.

[Hen96] T. Henzinger. The theory of hybrid automata. *Proceedings of the 11th Annual Symposium on Logic in Computer Science, LNCS*, pages 278–292, 1996.

[HHWT95] T. Henzinger, P.H. Ho, and H. Wong-Toi. Hytech : The next generation. *Proceeding of the 16th Annual Real Time Systems Symposium*, pages 56–65, 1995.

[HKPV95] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata ? *Proceeding of the 27th Annual Symposium on Theory of Computing*, pages 373–382, 1995.

[Ho95] P.H. Ho. *Automatic Analysis of Hybrid Systems.* PhD thesis, Cornell University, 1995.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Langages and Computation.* Addison-Wesley, 1979.

[JAGZ00] G. Juanole, S. Abdelatif, L. Gallon, and M. Adnene Zayene. Nouveaux concepts sur les transitions du modèle rdpts : priorités dynamiques et attributs temporels dynamiques. *Conference Internationale Francophone d'Automatique, CIFA 2000*, pages 976–981, 2000.

[Kha97] W. Khansa. *Réseaux de Petri P-temporels : Contribution à l'étude des systèmes à événements discrets.* PhD thesis, Université de Savoie, Annecy, France, 1997.

[Kou99] N. El Kouhen. *Commande supervisée des systèmes à événements discrets temporisés.* PhD thesis, Laboratoire d'Automatique et Automatique Industrielle - Ecole Mohammadia d'Ingénieurs, Rabat, 1999.

[Kum91] R. Kumar. *Supervisory Synthesis Techniques for Discrete Event Dynamical Systems.* PhD thesis, University of Texas AT Austin, 1991.

[MF76] P. Merlin and D.J. Faber. Recoverability of communication protocols. *IEEE Transaction on Communication*, COM-24(9), 1976.

[MPS95] O. Maler, A. Puneli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. *Proc. STACS'95, LNCS*, 900 :229–242, 1995.

[Mur89] T. Murata. Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, 77(4) :541–580, 1989.

[Oli94] A. Olivero. *Modélisation et analyse des systèmes temporisés et hybrides*. PhD thesis, VERIMAG - Institut National Polytechnique de Grenoble, 1994.

[PV94] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. *In D.L.Dill editor, CAV'94, Computer-aided Verification, LNCS*, 818 :85–104, 1994.

[RW87] J.G. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J., Control and Optimisation*, 25 :206–230, 1987.

[RW89] J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1) :81–97, 1989.

[SA] A.T. Sava and H. Alla. Du réseaux de petri t-temporel à l'automate temporisé. *soumis au Journal Européen des Systèmes Automatisés, en révision.*

[SA01a] A.T. Sava and H. Alla. Combining hybrid petri nets and hybrid automata. *à apparaitre dans IEEE Transactions on Robotics and Automation*, October 2001.

[SA01b] A.T. Sava and H. Alla. Commande par supervision des systèmes à événements discrets temporisés. *Accepté pour présentation au Colloque Francophone sur la Modélisation des Systèmes Réactifs*, octobre 2001.

[SAFG01] A.T. Sava, H. Alla, J.L. Ferrier, and A. Gouin. Du réseaux de petri t-temporel à l'automate temporisé en vue de la commande par supervision. *Journées Nationales de l'Automatique*, février 2001.

[Sav00] A.T. Sava. Automate temporisé et graphe des classes d'états, outils pour l'analyse des rdp t-temporels. *Conference Internationale Francophone d'Automatique, CIFA 2000*, pages 785–789, 2000.

[WR87] W.M. Wonham and J.G. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J., Control and Optimisation*, 25(3) :637–659, 1987.

[Yov93] S. Yovine. *Méthodes et outils pour la vérification symbolique des systèmes temporisés*. PhD thesis, VERIMAG - Institut National Polytechnique de Grenoble, 1993.

[Yov97] S. Yovine. Kronos : A verification tool for real-time systems. *Springer International Journal of Software Tools for Technology Transfer*, 1(1/2), 1997.

[Yov98] S. Yovine. Model checking timed automata. *Embedded Systems, G. Rosenberg and F. Vaandrager eds., LNCS*, 1494, 1998.

Annexe A

Illustration de l'algorithme de synthèse de la commande

Dans cette annexe nous illustrons le déroulement de l'algorithme que nous avons proposé pour la synthèse de la commande des SEDT.

L'exemple considéré est le poste de collage présenté dans la section 2.1.5. Dans le chapitre 3 nous avons construit l'automate temporisé associé au RdP T-temporel modélisant ce procédé. L'automate obtenu est illustré dans la figure 3.10.

Pour améliorer la présentation, nous reprenons cet automate dans la figure A.1.

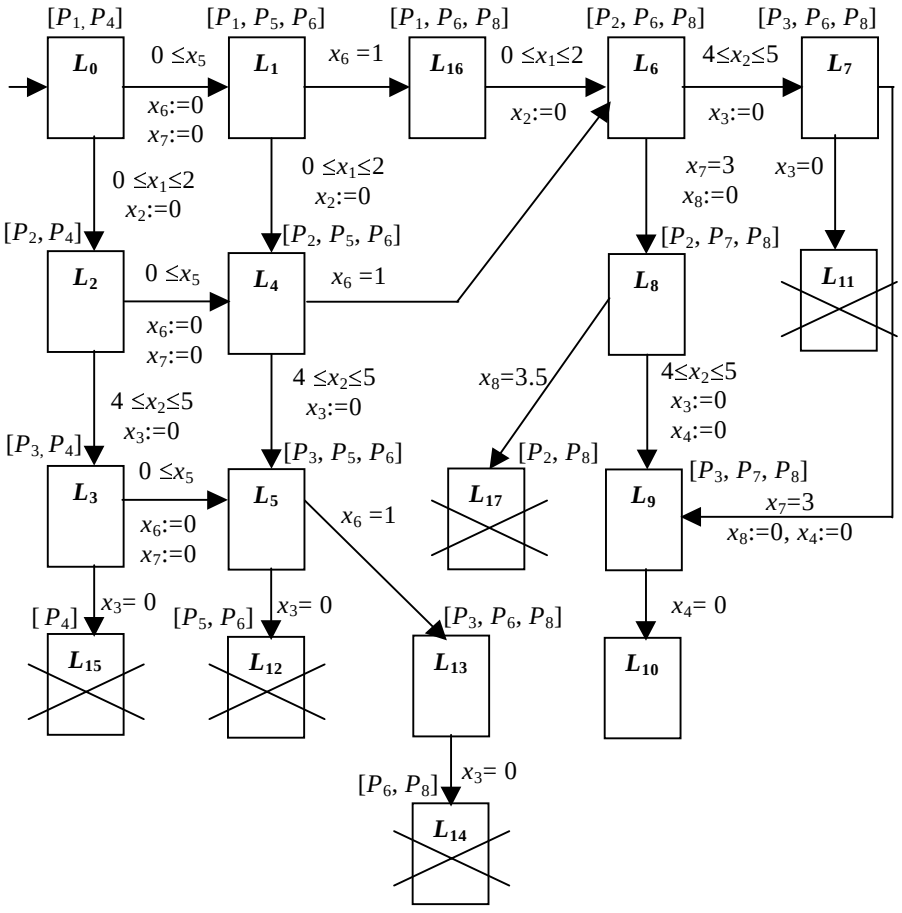


FIG. A.1 – Automate temporisé du poste de collage

Nous rappelons que les transitions contrôlables de cet automate sont $T_{0,1}$, $T_{0,2}$, $T_{1,4}$,

$T_{2,4}$ et $T_{16,6}$.

Les sommets interdits sont L_{11}, L_{12}, L_{14}, L_{15} et L_{17}.

La synthèse de la commande pour ce système consiste à calculer des nouvelles gardes pour les transitions $T_{0,1}$, $T_{0,2}$, $T_{1,4}$, $T_{2,4}$ et $T_{16,6}$ telles que les sommets interdits ne soient jamais atteignables.

Avant de commencer l'exécution de l'algorithme, on initialise l'ensemble des sommets interdits :

$$F := \{L_{11}, L_{12}, L_{14}, L_{15}, L_{17}\}.$$

Pas 1 : *Initialisation*

1.1. Lorsque $F \neq \phi$ on continue avec le **pas 1.2.**

1.2. Soit L_{14} un sommet interdit.

1.3. Soit L_{13} un sommet à partir duquel on peut atteindre le sommet interdit L_{14} par le franchissement d'une transition.

Le sommet L_{13} n'a pas d'autre transition de sortie. Ainsi, la seule évolution possible depuis ce sommet est de franchir la transition $T_{13,14}$ vers L_{14}. Dans ce cas L_{13} devient lui même un sommet interdit :

- On élimine de l'automate la transition $T_{13,14}$ et le sommet L_{14}.
- On actualise l'ensemble F :
  - On ajoute le sommet L_{13} à l'ensemble F :

$$\begin{aligned} F &:= F \cup \{L_{13}\} \\ &:= \{L_{11}, L_{12}, L_{13}, L_{15}, L_{17}\} \end{aligned}$$

- **Allez au pas 1**

Pas 1 : *Initialisation*

1.1. Lorsque $F \neq \phi$ on continue avec le **pas 1.2.**

1.2. Soit L_{13} un sommet interdit.

1.3. Soit L_5 un sommet à partir duquel on peut atteindre le sommet interdit L_{13} par le franchissement d'une transition.

Le sommet L_5 n'a aucune transition de sortie qui mène vers un sommet non-interdit. Il y a deux possibilités d'évolution depuis L_5 :

1) on franchit la transition $T_{5,12}$ vers le sommet interdit L_{12} ;

2) on franchit la transition $T_{5,13}$ vers le sommet interdit L_{13}.

Par conséquent L_5 devient lui même un sommet interdit :

- On élimine de l'automate les transitions $T_{5,12}$ et $T_{5,13}$, ainsi que les sommets L_{12} et L_{13}.
- On actualise l'ensemble F :
  - On ajoute L_5 à l'ensemble F

$$\begin{aligned} F &:= F \cup \{L_5\} \\ &:= \{L_5, L_{11}, L_{12}, L_{13}, L_{15}, L_{17}\} \end{aligned}$$

  - On enlève les sommets L_{12} et L_{13} de l'ensemble F :

$$F := \{L_5, L_{11}, L_{15}, L_{17}\}$$

- **Allez au pas 1**

L'automate obtenu après avoir analysé les sommets interdits L_{13} et L_{14} est illustré dans la figure A.2

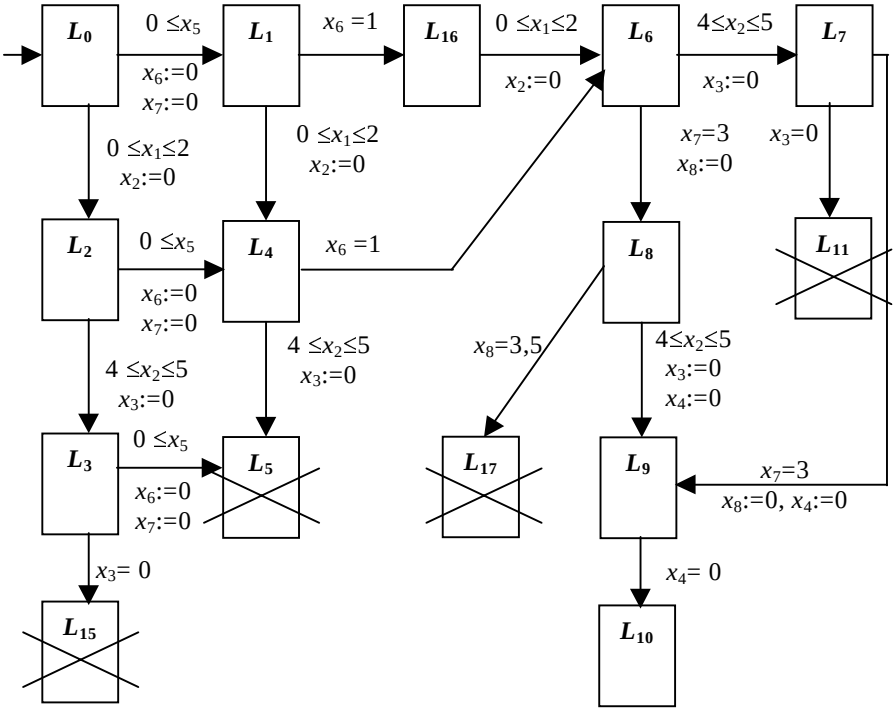


FIG. A.2 – Automate obtenu après l'analyse des sommets L_{13} et L_{14}

Pas 1 : *Initialisation*

1.1. Lorsque $F \neq \phi$ on continue avec le **pas 1.2.**

1.2. Soit L_5 un sommet interdit.

1.3. Soit L_4 un sommet à partir duquel on peut atteindre le sommet interdit L_5 par le franchissement d'une transition. Ce sommet a une transition de sortie $T_{4,6}$ vers le sommet non-interdit L_6. Alors, on initialise la pile P et l'ensemble des horloges à partir duquel il faut actualiser l'automate.

- $Q := \phi$
- $P := \phi$

Pas 2 : *Traitement aval*

2.1. On calcule l'espace des horloges à l'entrée dans le sommet L_4 :

Le sommet L_4 a deux transitions d'entrée : $T_{1,4}$ et $T_{2,4}$.

L'espace des horloges à l'entrée dans L_4 par le franchissement de la transition $T_{1,4}$, noté $E_{1,4}$, est connu. Il a été calculé lors de la construction de l'automate. Supposons que cet espace est décrit par l'expression suivante :

$$E_{1,4} = [0 \leq x_1 - x_6 \leq 2 \wedge 0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0]$$

L'espace des horloges à l'entrée dans L_4 par le franchissement de la transition $T_{2,4}$, noté $E_{2,4}$, est également connu. Supposons qu'il est décrit par l'expression suivante :

$$E_{2,4} = [0 \leq x_2 - x_5 \leq 2 \wedge 0 \leq x_5 \leq 5 \wedge x_6 = 0 \wedge x_7 = 0]$$

L'espace des horloges à l'entrée dans L_4 est :

$$\begin{aligned} E_4 &= E_{1,4} \vee E_{2,4} \\ &= [0 \leq x_1 - x_6 \leq 2 \wedge 0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0] \vee \\ &\quad [0 \leq x_2 - x_5 \leq 2 \wedge 0 \leq x_2 \leq 5 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

2.2. On calcule l'espace des horloges actives dans le sommet L_4 :

- D'abord on calcule l'espace des horloges actives à l'entrée dans ce sommet. Les horloges actives dans le sommet L_4 sont x_2, x_6 et x_7. Cette information a été mémorisée lors de la construction de l'automate

$$\begin{aligned} E_4^a &= Pr_{x_2,x_6,x_7}(E_4) \\ &= [0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0] \vee \\ &\quad [0 \leq x_2 \leq 5 \wedge x_6 \wedge x_7 = 0] \end{aligned}$$

- Ensuite, on calcule l'espace des horloges actives dans L_4. Cet espace est le successeur continu de l'espace E_4^a. Le calcul de cet espace nécessite la connaissance de l'invariant du sommet L_4. Ceci a été calculé lors de la construction de l'automate temporisé.
  L'invariant du sommet L_2 est :

$$I(L_4) = [0 \leq x_2 \leq 5 \wedge 0 \leq x_6 \leq 1 \wedge 0 \leq x_7 \leq 3]$$

  L'espace des horloges actives dans L_4 est :

$$\begin{aligned} Suc_t(E_4^a) &= \exists t \in \mathbb{R}^+ . E_4^a[x_2 - t, x_6 - t, x_7 - t].I(L_4) \\ &= [0 \leq x_6 - x_2 \leq 1 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge 0 \leq x_2 \leq 5] \vee \\ &\quad [0 \leq x_2 - x_6 \leq 5 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge 0 \leq x_2 \leq 5] \end{aligned}$$

2.3. La garde de la transition $T_{4,5}$ vers le sommet interdit L_5 est :

$$g_{4,5} = 4 \leq x_2 \leq 5$$

Le sommet L_4 a une seule transition vers un sommet non-interdit. Cette transition est $T_{4,6}$.
On calcule l'espace des horloges $(E_4^d)_{4,6}$ désiré à l'entrée dans L_5. Cet espace mémorise les valuations des horloges à l'entrée dans L_5 telles que la transition $T_{5,6}$ est toujours franchie avant $T_{4,5}$.
Puisque $T_{4,6}$ est incontrôlable, faire :
La garde de $T_{4,6}$ est :

$$x_6 = 1$$

- Calculer la condition sur la valeur des horloges actives dans L_4 :

$$4 - x_2 > 1 - x_6$$

- Calculer l'espace des horloges actives désiré $(D_4)_{4,6}$:

$$\begin{aligned} (D_4)_{4,6} &= Suc_t(E_4^a) \wedge [4 - x_2 > 1 - x_6] \\ &= Suc_t(E_4^a) \wedge [x_2 - x_6 < 3] \\ &= [0 \leq x_6 - x_2 \leq 1 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge 0 \leq x_2 \leq 5] \vee \\ &\quad [0 \leq x_2 - x_6 < 3 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge 0 \leq x_2 \leq 5] \end{aligned}$$

- On calcule l'espace des horloges désiré à l'entrée dans L_4 :

$$\begin{aligned} (E_4^d)_{4,6} &= E_4 \wedge (D_4)_{4,6} \\ &= [0 \leq x_1 - x_6 \leq 2 \wedge 0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0] \vee \\ &\quad [0 \leq x_2 - x_5 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

2.4. On calcule l'espace des horloges désiré à l'entrée dans L_4

$$\begin{aligned} E_4^d &= (E_4^d)_{4,6} \\ &= [0 \leq x_1 - x_6 \leq 2 \wedge 0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0] \vee \\ &\quad [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

Puisque $\mathbf{E_4^d \neq \phi}$ et $\mathbf{E_4^d \neq E_4}$, il faut remonter dans l'automate.

- On met à jour l'ensemble des sommets à partir duquel il faut actualiser l'automate :

$$Q := \{L_4\}$$

- On mémorise dans la pile P le sommet L_4, les horloges actives dans ce sommet et l'espace E_4^d :

$$P := \{[L_4; x_2, x_6, x_7; E_4^d]\}$$

Pas 3 : *Traitement amont*

On analyse le dernier élément introduit dans la pile. Cet élément mémorise le sommet L_4, les horloges actives dans ce sommet (x_2, x_6 et x_7) et l'espace E_4^d.

3.1. On enlève cet élément de la pile :

$$P := \phi$$

3.2. On détermine les transitions d'entrée dans L_4 qu'on doit remonter.
Le sommet L_4 a deux transitions d'entrée : $T_{1,4}$ et$T_{2,4}$.
Analyser la transition $T_{1,4}$:
On calcule l'espace des horloges désiré à l'entrée dans L_4 par le franchissement de $T_{1,4}$:

$$\begin{aligned} E_{1,4}^d &= E_{1,4} \wedge E_4^d \\ &= [0 \leq x_1 - x_6 \leq 1 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge 0 \leq x_2 \leq 5] \\ &= E_{1,4} \end{aligned}$$

Puisque $E_{1,4} \subseteq E_4^d$ il n'est pas nécessaire de remonter la transition $T_{1,4}$.
Analyser la transition $T_{2,4}$:
On calcule l'espace des horloges désiré à l'entrée dans L_4 par le franchissement de $T_{2,4}$:

$$\begin{aligned} E_{2,4}^d &= E_{2,4} \wedge E_4^d \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

Il faut remonter la transition $T_{2,4}$ puisque $E_{2,4} \not\subset E_4^d$.

3.3. Il y a une seule transition à remonter, $T_{2,4}$. Pour cette transition on effectue les opérations suivantes :

- On calcule l'espace des horloges à l'entrée dans le sommet source de $T_{2,4}$, i.e. L_2. Ce sommet a une seule transition d'entrée, $T_{0,2}$. L'espace des horloges à l'entrée dans L_2 par le franchissement de $T_{2,4}$ est connu. Il a été calculé lors de la construction de l'automate temporisé.

$$E_{0,2} = [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0]$$

L'espace des horloges à l'entrée dans le sommet L_2 est :

$$\begin{aligned} E_2 &= E_{0,2} \\ &= [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0] \end{aligned}$$

- On calcule l'espace des horloges actives dans le sommet L_2.
  - On calcule l'espace des horloges actives à son entrée. Les horloges actives dans le sommet L_2 sont x_2 et x_5. Cette information a été mémorisée lors de la construction de l'automate

$$\begin{aligned} E_2^a &= Pr_{x_2,x_5}(E_2) \\ &= [0 \leq x_5 \leq 2 \wedge x_2 = 0] \end{aligned}$$

  - On calcule l'espace des horloges actives dans L_2. Cet espace est le successeur continu de l'espace E_2^a. Le calcul de cet espace nécessite la connaissance de l'invariant du sommet L_2. Ceci a été calculé lors de la construction de l'automate temporisé.
    L'invariant du sommet L_2 est :

$$I(L_2) = [0 \leq x_2 \leq 5]$$

    L'espace des horloges actives dans L_2 est :

$$\begin{aligned} Suc_t(E_2^a) &= \exists t \in \mathbb{R}^+ . E_2^a[x_2 - t, x_5 - t].I(L_2) \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 \leq 5] \end{aligned}$$

Puisque $T_{2,4}$ est contrôlable, faire :

- On calcule la nouvelle garde $g_{2,4}^d$
  - On calcule l'espace $(E_{2,4}^d)'$. Cet espace est obtenu par la projection de l'espace $E_{2,4}^d$ sur les dimensions des horloges actives dans L_2 qui ne sont pas mises à zéro par l'affectation associée à la transition $T_{2,4}$. Les horloges qui ont cette propriété sont x_2 et x_5.

$$\begin{aligned} (E_{2,4}^d)' &= Pr_{x_2,x_5}(E_{2,4}^d) \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

  - Calculer la nouvelle garde $g_{2,4}^d$:

$$\begin{aligned} g_{2,4}^d &= g_{2,4} \wedge (E_{2,4}^d)' \\ &= [0 \leq x_5] \wedge [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

- Calculer l'espace des horloges désiré à l'entrée dans L_2 :
  - On calcule l'ensemble des valuations des horloges dans L_2 qui vérifient la garde $g_{2,4}^d$:

$$\begin{aligned} S_{2,4}^d &= Suc_t(E_2^a) \wedge g_{2,4}^d \\ &= [0 \leq x_5 \leq 2 \wedge 0 \leq x_2 \leq 5] \wedge \\ &\quad [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

  - On calcule l'espace des horloges actives désiré dans L_2. L'ensemble des valuations qu'on souhaite atteindre pendant le séjour du système dans ce sommet est décrit par le prédécesseur continu de l'espace $S_{2,4}^d$.

$$\begin{aligned} Pre_t(S_{2,4}^d) &= \exists t \in \mathbb{R}^+ . S_{2,4}^d[x_2 + t, x_5 + t] \\ &= \exists t \in \mathbb{R}^+ . [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 + t < 3] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

L'espace des horloges actives désiré dans L_2 est :

$$\begin{aligned} D_2 &= Suc_t(E_2^a) \wedge Pre_t(S_{2,4}^d) \\ &= [0 \leq x_5 \leq 2 \wedge 0 \leq x_2 \leq 5] \wedge \\ &\quad [0 \leq x_5 \leq 2 \wedge 0 \leq x_2 < 3] \end{aligned}$$

- On calcule l'espace des horloges désiré à l'entrée dans le sommet L_2 :

$$\begin{aligned} E_2^d &= E_2 \wedge D_2 \\ &= [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0] \wedge \\ &\quad \wedge [0 \leq x_5 \leq 2 \wedge 0 \leq x_2 < 3] \\ &= [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0] \end{aligned}$$

  Puisque $\mathbf{E_2^d = E_2}$ alors il n'est pas nécessaire de remonter au-delà du sommet L_2.
- On met à jour l'invariant du sommet L_2.
  - On détermine la valeur maximale des horloges x_2 et x_5 dans le sommet L_2 :

$$\begin{aligned} (x_2)_{max} &= max\{Pr_{x_2}(D_2)\} \\ &= max\{0 \leq x_2 < 3\} \end{aligned}$$

$$\begin{aligned} (x_5)_{max} &= max\{Pr_{x_5}(D_2)\} \\ &= max\{0 \leq x_5 < 5\} \end{aligned}$$

  - Le nouvel invariant du L_2 est :

$$\begin{aligned} I(L_2)^d &= I(L_2) \wedge [0 \leq x_2 < 3] \wedge [0 \leq x_5 < 5] \\ &:= [0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5] \end{aligned}$$

- On met à jour l'ensemble Q :

$$\begin{aligned} Q &:= Q \setminus \{L_4\} \\ &:= Q \cup \{L_2\} \\ &:= \{L_2\} \end{aligned}$$

3.4. Lorsque $P = \phi$ cette étape de la synthèse de la commande concernant le sommet interdit L_5 est terminée. Nous avons trouvé une nouvelle garde pour la transition $T_{2,4}$ telle que le sommet interdit L_5 ne soit jamais atteignable depuis L_4. Par la suite il faut actualiser l'automate pour prendre en compte les éventuels effets la modification de la garde de $T_{2,4}$.

Pas 4 : *Actualiser l'automate*

L'ensemble des sommets à partir desquels il faut actualiser l'automate est :

$$Q = \{L_2\}$$

4.1. Considérons le sommet L_2 de l'ensemble Q.

- On enlève le sommet L_2 de l'ensemble Q :

$$\begin{aligned} Q &:= Q \setminus \{L_2\} \\ &:= \phi \end{aligned}$$

- On calcule l'espace des horloges à l'entrée dans le sommet L_2. Ce sommet a une seule transition d'entrée, $T_{0,2}$. L'espace des horloges à l'entrée dans L_2 par le franchissement de $T_{2,4}$ a été calculé lors de la construction de l'automate temporisé.

$$E_{0,2} = [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0]$$

L'espace des horloges à l'entrée dans le sommet L_2 est :

$$\begin{aligned} E_2 &= E_{0,2} \\ &= [0 \leq x_1 \leq 2 \wedge x_1 = x_5 \wedge x_2 = 0] \end{aligned}$$

- On mémorise dans la pile P le sommet L_2, les horloges actives dans ce sommet (x_2 et x_5) et l'espace E_2.

$$P := \{[L_2; x_2, x_5; E_2]\}$$

4.2. On analyse le dernier élément introduit dans la pile P. Il s'agit du sommet L_2 avec l'espace des horloges à son entrée E_2.

- Enlever cet élément de la pile :

$$\begin{aligned} P &:= P \setminus \{[L_2; x_2, x_5; E_4]\} \\ &:= \phi \end{aligned}$$

- On calcule l'espace des horloges actives dans le sommet L_2.
  - On calcule l'espace des horloges actives à son entrée. Les horloges actives dans le sommet L_2 sont x_2 et x_5. Cette information a été mémorisée lors de la construction de l'automate

$$\begin{aligned} E_2^a &= Pr_{x_2,x_5}(E_2) \\ &= [0 \leq x_5 \leq 2 \wedge x_2 = 0] \end{aligned}$$

  - On calcule l'espace des horloges actives dans L_2. Cet espace est le successeur continu de l'espace E_2^a. Le calcul de cet espace nécessite la connaissance de l'invariant du sommet L_2. L'invariant de ce sommet a été actualisé dans l'étape traitement amont.

$$I(L_2) = [0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5]$$

L'espace des horloges actives dans L_2 est :

$$\begin{aligned} Suc_t(E_2^a) &= \exists t \in \mathbb{R}^+ . E_2^a[x_2 - t, x_5 - t].I(L_2) \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5] \end{aligned}$$

4.3. On détermine les transitions de sortie de L_2 qui ne sont plus franchissables. Le sommet L_2 a deux transitions de sortie : $T_{2,3}$ et $T_{2,4}$.

Analyser la transition $T_{2,3}$ La garde de cette transition est $g_{2,3} = [4 \leq x_2 \leq 5]$

- On calcule l'ensemble des valuations des horloges dans L_2 qui satisfont la garde de cette transition :

$$\begin{aligned} S_{2,3} &= Suc_t(E_2^a) \wedge g_{2,3} \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5] \wedge [4 \leq x_2 \leq 5] \\ &= \phi \end{aligned}$$

Par conséquent, la transition $T_{2,3}$ ne peut plus être franchie depuis L_2.

- On supprime cette transition de l'automate.
- On supprime également le sommet L_3, les transitions $T_{3,5}$ et $T_{3,15}$, ainsi que le sommet L_{15}. On enlève le sommet L_{15} de l'ensemble F :

$$\begin{aligned} F &:= F \setminus \{L_{15}\} \\ &:= \{L_{11}, L_{17}, L_5\} \end{aligned}$$

Analyser la transition $T_{2,4}$ La garde de cette transition est $g_{2,4} = [0 \leq x_5 \wedge x_3 < 3]$

- On calcule l'ensemble des valuations des horloges dans L_2 qui satisfont la garde de cette transition :

$$\begin{aligned} S_{2,4} &= Suc_t(E_2^a) \wedge g_{2,4} \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5] \wedge [0 \leq x_5 \wedge x_3 < 3] \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5] \\ &\neq \phi \end{aligned}$$

Par conséquent, la transition $T_{2,4}$ reste franchissable.

- On calcule l'espace des horloges à l'entrée dans le sommet L_2 par le franchissement de $T_{2,4}$:

$$\begin{aligned} e_{2,4} &= Suc_{2,4}(S_{2,4}) \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

- On actualise l'espace des horloges à l'entrée dans L_2 par le franchissement de $T_{2,4}$. Lorsqu'on visite ce sommet pour la première fois pendant cette étape, alors :

$$\begin{aligned} E_{2,4} &= e_{2,4} \\ &= [0 \leq x_5 - x_2 \leq 2 \wedge 0 \leq x_2 < 3 \wedge 0 \leq x_5 < 5 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

- On calcule l'espace des horloges à l'entrée dans le sommet L_4 :

$$\begin{aligned} E_4 &= E_{1,4} \vee E_{2,4} \\ &= [0 \leq x_1 - x_6 \leq 2 \wedge 0 \leq x_1 \leq 2 \wedge 0 \leq x_6 \leq 1 \wedge x_6 = x_7 \wedge x_2 = 0] \vee \\ &\quad [0 \leq x_2 - x_5 \leq 2 \wedge 0 \leq x_2 < 3 \wedge x_6 = 0 \wedge x_7 = 0] \end{aligned}$$

Le sommet L_4 a deux transitions de sortie. Alors il faut continuer l'actualisation de l'automate depuis ce sommet. On mémorise dans la pile P le sommet L_4, les horloges actives dans ce sommet (x_2, x_6 et x_7) et l'espace E_4.

4.4. Lorsque $P \neq \phi$ **Allez au pas 4.2.**

L'automate temporisé obtenu après cette première itération de la procédure d'actualisation de l'automate est illustré dans la figure A.3.

Le sommet interdit L_5 sera lui aussi supprimé lors de l'itération concernant le sommet L_4.

En continuant l'exécution de l'algorithme proposé pour la synthèse de la commande on obtient l'automate temporisé présenté dans la figure A.4.

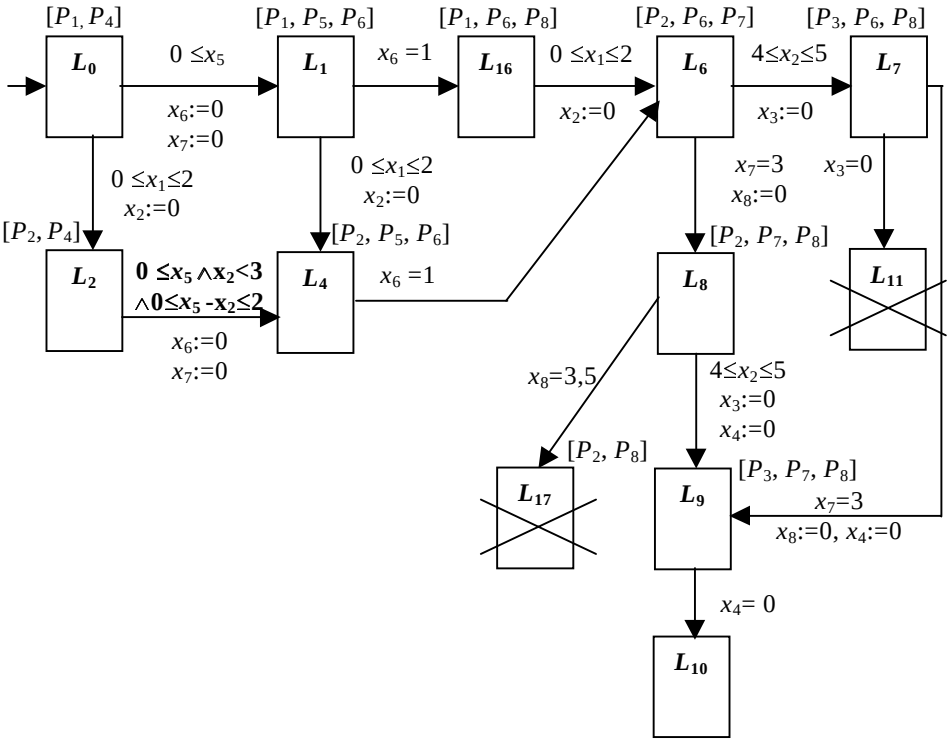


FIG. A.3 – Automate obtenu après la première itération

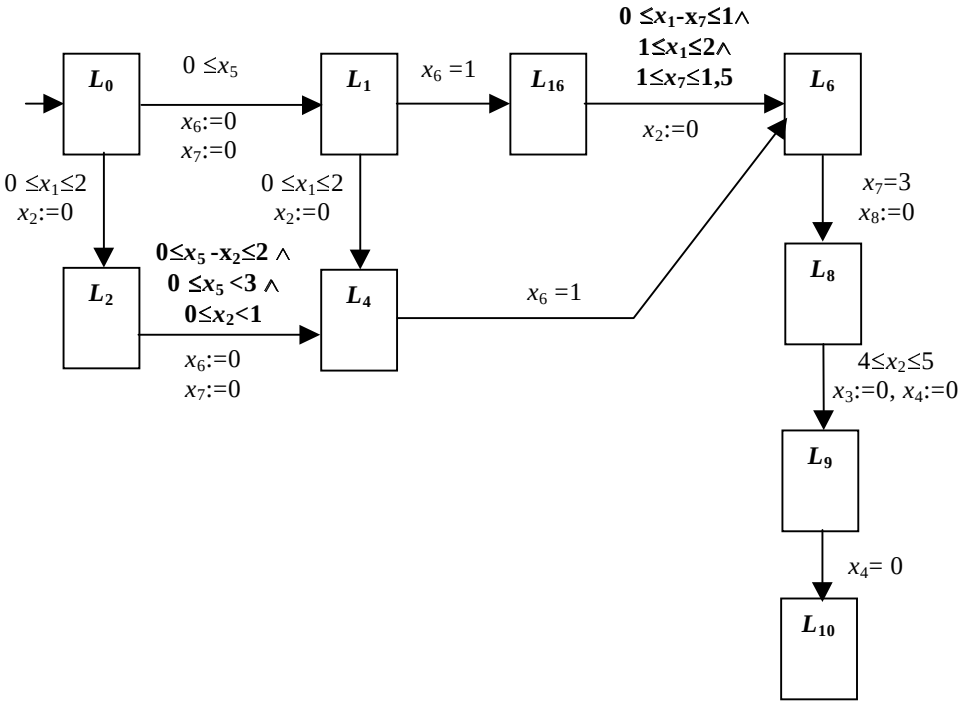


FIG. A.4 – Modèle de commande du poste de collage