ÉPREUVE COMMUNE DE TIPE - Partie D

TITRE :

Composition et décomposition d'automates temporisés

Temps de préparation : ……………..….2 h 15 minutes
Temps de présentation devant le jury : …….10 minutes
Entretien avec le jury : ……………………..10 minutes

GUIDE POUR LE CANDIDAT :

Le dossier ci-joint comporte au total : *11* pages

Document principal (10 pages) : composition et décomposition des automates temporisés
Documents complémentaires (1 pages) : Annexe - exemple

Travail **suggéré** au candidat :

- Décrire le principe de fonctionnement des automates à contrainte de temps ainsi que le principe de la composition et de la décomposition de ces automates.
- Mettre en évidence, en s'appuyant sur les exemples donnés et, au besoin, en présentant d'autres exemples, l'intérêt et les limites des automates à contrainte de temps. Proposer d'autres domaines où de tels automates pourraient s'appliquer.

CONSEILS GENERAUX POUR LA PREPARATION DE L'EPREUVE :

* Lisez le dossier en entier dans un temps raisonnable.
* Réservez du temps pour préparer l'exposé devant le jury.

- Vous pouvez écrire sur le présent dossier, le surligner, le découper… mais tout sera à remettre au jury en fin d'oral.
- En fin de préparation, rassemblez et ordonnez soigneusement TOUS les documents (transparents, *etc.*) dont vous comptez vous servir pendant l'oral, ainsi que le dossier, les transparents et les brouillons utilisés pendant la préparation. En entrant dans la salle d'oral, vous devez être prêts à débuter votre exposé.
- A la fin de l'oral, vous devez remettre au jury le présent dossier, les transparents et les brouillons utilisés pour cette partie de l'oral, ainsi que TOUS les transparents et autres documents présentés pendant votre prestation.

COMPOSITION ET DECOMPOSITION D’AUTOMATES TEMPORISES

1. Introduction

Les automates à états finis sont très couramment utilisés pour modéliser les systèmes réactifs. Ce type de modèles permet d'étudier l'ordre relatif des actions, comme « déclencher l'alarme après toute réception d'un problème ». Mais il ne contient pas d'information quantitative sur le délai séparant deux actions consécutives. Les automates temporisés, proposé par Alur et Dill, sont une extension des automates finis classiques qui décrivent des comportements, mais à chaque action est ajouté une date, qui correspond à l'instant auquel elle a lieu. A chaque automate est associé un ensemble fini d'horloges, ces horloges étant des variables qui évoluent toutes à la même vitesse (celle d'une horloge universelle externe, sur laquelle on ne peut pas agir) et pouvant être remises à zéro. On peut mettre des contraintes sur les transitions de l'automate, comme par exemple : « pour pouvoir effectuer cette transition, il faut que l'horloge x ait sa valeur comprise entre 2 et 3 ».

Lorsque l’on étudie des systèmes réels, les modèles qui correspondent sont souvent très gros. Il est alors judicieux de les modéliser non plus par un gros automate, mais par un réseau d'automates plus petits (synchronisés entre eux). On parle alors de décomposition des automates en *modules* de taille plus petite. Les propriétés que l'on cherche à étudier (au niveau du gros système) doivent alors être liées aux propriétés des petits systèmes.

2. Automate temporisé et générateurs contraints

2.1. mots temporisés et horloges

Soit Z un ensemble. Notons Z^+ (respectivement Z^ω) l'ensemble de séquences finies non vides (respectivement infinies) d’éléments de Z, avec ε pour la séquence vide, et $Z^\infty = Z^+ \cup Z^\omega$ l'ensemble de toutes les séquences non vides d’éléments de Z.

Nous considérerons un domaine de temps $\boldsymbol{T}$ qui est l'ensemble des nombres entiers non négatifs $\mathbb{N}$ ou l'ensemble des nombres rationnels non négatifs $\mathbb{Q}_+$. Une séquence de dates sur $\boldsymbol{T}$ est une suite finie ou infinie de dates $\tau = (t_i)_{i\geq 1}$ d’éléments de $\boldsymbol{T}$. Les actions des automates considérés sont données par un ensemble fini Σ. Un mot temporisé est un élément de $(\Sigma\times\boldsymbol{T})^\infty$, il peut être vu comme un couple $w = (\sigma,\tau)$, où $\sigma = (a_i)_{i\geq 1}$ un mot dans Σ^∞ et $\tau = (t_i)_{i\geq 1}$ une séquence de dates dans $\boldsymbol{T}^\infty$ de même longueur que σ. Par exemple, (a, 1.5)(b, 2)(a, π)(b, 3.9)(a, 3.9) est un mot temporisé fini, tandis que $((ab)^\omega,(i + 1/i)_{i\geq 1})$ est un mot temporisé infini (avec pour $m \in Z^+$, m^ω est la répétition infinie de la séquence m).

La concaténation d'un mot temporisé fini u = $(a_i,t_i)_{1\leq i\leq n}$ et d'un mot temporisé fini ou infini v = $(a'_i,t'_i)_{1\leq i}$ est définie si seulement si $t_n \leq t'_1$ et est égal au mot temporisé u . v = $(a''_i,t''_i)_{1\leq i}$ avec $(a''_i,t''_i) = (a_i,t_i)$ si $1 \leq i \leq n$ et $(a''_i,t''_i) = (a'_{i-n},t'_{i-n})$ si $n < i$.

Soit X un ensemble de variables typées dans $\boldsymbol{T}$, appelées horloges. Parmi les éléments de X, nous distinguons une horloge spéciale x_0 qui dénotera le temps absolu.

A chaque transition de l'automate on associe un prédicat sur la valeur des horloges, appelé garde. Les gardes (ou contraintes de temps) sur X sont les formules définies comme suit :

- $\top$ est une garde (pas de contrainte)
- $x \sim c$ et $(x - y) \sim c$ sont des gardes (avec x et y des horloges, $c \in \boldsymbol{T}$ et $\sim$ est un opérateur binaire parmi $\{ <, =, > \}$)
- $g_1 \vee g_2$, $g_1 \wedge g_2$ et $\neg g_1$ sont des gardes ssi g_1 et g_2 sont des gardes.

L'ensemble des gardes de X est noté Guards(X).

Une valuation des horloges de X est une fonction $\alpha : X \rightarrow \boldsymbol{T}$ qui associe à chaque horloge x une valeur du temps $\alpha(x)$. L'ensemble de toutes les valuations d'horloge est noté $\boldsymbol{T}^X$. Si Y est un sous-ensemble de X, $\alpha[Y \leftarrow 0]$ désigne la valuation qui à $x \in Y$ associe 0 et à $x \notin Y$ associe $\alpha(x)$.

Pour α valuation d'horloges, la relation de satisfaction suivante est définie, g étant une garde :

$\alpha \models g \Leftrightarrow g(x_i/\alpha(x_i))$ est vraie où $g(x_i/\alpha(x_i))$ désigne la valeur booléenne de g dans lequel x_i est remplacé par sa valeur $\alpha(x_i)$.

2.2. Automates temporisés et exécution

Un automate temporisé de Büchi défini sur Σ, $\boldsymbol{T}$ et X est un quintuplet $A = (Q, T, I, F, R)$, où Q est un ensemble fini d'états, $T \subseteq Q \times [\text{Guards}(X) \times \Sigma \times \wp(X)] \times Q$ est l'ensemble des transitions (avec $\wp(X)$ l'ensemble des parties de X), $I \subseteq Q$ est le sous-ensemble des états initiaux, $F \subseteq Q$ est le sous-ensemble des états finaux et $R \subseteq Q$ est un sous-ensemble d'états répétés correspondant à une condition de Büchi, comme indiqué ci-après. Une transition a la forme (p, g, a, r, q), où g est une garde, a une action de Σ et $r \subseteq X$ le sous-ensemble des horloges à réinitialiser. Elle signifie que si le système se trouve dans l'état p et que la garde g est vérifiée, alors le système peut effectuer l'action a, mettre les horloges de r à zéro et aller dans l'état q.

Nous supposons que l'horloge particulière x_0 n'est jamais réinitialisée.

L'ensemble de tous les automates temporisés définis sur l'ensemble d'actions Σ, le domaine temporel $\boldsymbol{T}$ et l'ensemble des horloges X est noté TA(Σ, $\boldsymbol{T}$, X).

Afin de définir comment un mot temporisé est accepté par un automate temporisé, précisons les notions de chemin et de parcours de chemin temporisé. Un chemin dans A est une séquence non vide, finie ou infinie, de transitions consécutives :

$$P = q_0 \xrightarrow{g_1, a_1, r_1} q_1 \xrightarrow{g_2, a_2, r_2} q_2 \ldots$$

où $(q_{i\text{-}1}, g_i, a_i, r_i, q_i) \in T$ pour tout $i > 0$. Le chemin P est valide (ou acceptable) s'il commence dans un état initial et, soit termine dans un état final, soit est infini et respecte une condition de Büchi : $inf(P) \cap R$ est non vide, où $inf(P)$ est l'ensemble d'états qui se produisent infiniment souvent dans P.

Une exécution de l'automate selon un chemin P et pour une valuation d'horloge initiale $\alpha \in \boldsymbol{T}^{X}$ est un triplet $R = (\alpha, P, (t_i)_{i>0})$ où $(t_i)_{i>0}$ est une séquence non décroissante de dates de même longueur que P ; t_i est l'instant où l'on a exécuté la transition $q_{i-1} \xrightarrow{g_i,\, a_i,\, r_i} q_i$. Intuitivement, cette transition peut être exécutée, seulement si la garde g_i est satisfaite par la valuation de l'horloge courante, et à ce moment-là les horloges de r_i sont réinitialisées. Plus précisément, la valeur de l'horloge x au temps t se note $x(t)$, de sorte que la valuation d'horloge au temps t puisse être écrite comme le n-uplet $X(t) = (x(t))_{x \in X}$. En commençant au temps $t_0 = \alpha(x_0)$ et toute horloge x étant initialement valuée à $x(t_0) = \alpha(x)$, la valuation d'horloge au temps t est complètement déterminé par l'exécution : pour tout i $\geqslant$ 1,

$$x(t_i) = \begin{cases} 0 & \text{si } x \in r_i \\ x(t_{i-1}) + t_i - t_{i-1} & \text{sinon} \end{cases}$$

et $x(t) = x(t_{i\text{-}1}) + t - t_{i\text{-}1}$ pour $t_{i\text{-}1} \leqslant t < t_i$. De plus, nous avons besoin que $X(t_{i\text{-}1}) + t_i - t_{i\text{-}1} \models g_i$ pour tout i $\geqslant$ 1.

A toute exécution finie (resp. infinie) nous pouvons naturellement associer un mot temporisé fini (resp. infini), désigné comme une étiquette de R par

$$\ell(R) = (a_1, t_1)(a_2, t_2)\ldots \in (\Sigma \times \boldsymbol{T})^{\infty}.$$

Si $\ell(R)$ est fini et de longueur n, nous définissons aussi la valuation d'horloge $\mathrm{Ends}(R) \in \boldsymbol{T}^{X}$ « à la fin de R » de la façon suivante :

$$\mathrm{Ends}(R)(x) = \begin{cases} 0 & \text{si } x \in r_n \\ x(t_n) & \text{sinon} \end{cases}$$

Puisque nous avons posé que x_0 n'est jamais réinitialisé, on a toujours $\mathrm{Ends}(R)(x_0) = t_n$.

2.3. Automate temporisé et générateurs contraints

Avec cette définition, nous n'associons pas seulement un langage temporisé (i.e. un ensemble de mots temporisés) à un automate temporisé $\mathscr{A}$ mais une famille de langages temporisés classés par les valuations initiales des horloges. Plus précisément, pour n'importe quelle valuation d'horloge $\alpha \in \boldsymbol{T}^X$, nous définissons

$$\mathrm{L}_{\mathscr{A}}(\alpha) = \{\ \ell(R) \mid R = (\alpha, P, (t_i)_{i>0}) \text{ pour tout chemin } P \text{ acceptable dans } \mathscr{A}\}$$

Par soucis de simplicité, nous noterons $\mathrm{L}_{\mathscr{A},+}(\alpha)$ pour $\mathrm{L}_{\mathscr{A}}(\alpha) \cap (\Sigma \times \boldsymbol{T})^+$ et $\mathrm{L}_{\mathscr{A},w}(\alpha)$ pour $\mathrm{L}_{\mathscr{A}}(\alpha) \cap (\Sigma \times \boldsymbol{T})^w$.

Remarque 1 : Il n'est pas difficile de prouver que pour tout automate temporisé $\mathscr{A}$ et toute valuation d'horloge α, il existe un automate temporisé $\mathscr{B}$ tel que $\mathrm{L}_{\mathscr{A}}(\alpha) = \mathrm{L}_{\mathscr{B}}(\mathbf{0})$, où $\mathbf{0}$ dénote la valuation d'horloge définie par $\mathbf{0}(x) = 0$ pour toute horloge x.

La fonction Ends définie sur l'ensemble des exécutions permet aussi de définir l'ensemble des valuations d'horloges possibles après reconnaissance d'un mot temporisé fini. Précisément, pour tout mot temporisé $u \in (\Sigma \times \boldsymbol{T})^+$ et toute valuation initiale d'horloge $\alpha \in \boldsymbol{T}^X$, nous posons

$$\mathrm{Ends}_{\mathscr{A}}(\alpha,u) = \{\mathrm{Ends}(R) \mid R = (\alpha, P, (t_i)_{i>0})$$
$$\text{pour les chemins possibles } P \text{ dans } \mathscr{A} \text{ tels que } \ell(R) = u\}$$

Notons que si $u \notin \mathrm{L}_{\mathscr{A},+}(\alpha)$, il en résulte que $\mathrm{Ends}_{\mathscr{A}}(\alpha,u) = \varnothing$. Cette nouvelle fonction $\mathrm{Ends}_{\mathscr{A}}$ peut-être vue comme une contrainte sur les mots temporisés de $\mathrm{L}_{\mathscr{A},+}(\alpha)$ dans la mesure où cette fonction permet de vérifier qu'un mot est dans le langage si et seulement si la valeur de la fonction appliquée au mot n'est pas vide.

Nous définirons maintenant la notion de générateur contraint comme une généralisation de ces fonctions $\mathrm{L}_{\mathscr{A}}$ et $\mathrm{Ends}_{\mathscr{A}}$. Cette notion sera un outil fondamental pour la suite.

Un générateur contraint est un couple $(\mathscr{G}, \Lambda)$ tel que

1. $\mathscr{G} : \boldsymbol{T}^X \to \wp((\Sigma \times \boldsymbol{T})^\infty)$
2. $\Lambda : \boldsymbol{T}^X \times (\Sigma \times \boldsymbol{T})^+ \to \wp(\boldsymbol{T}^X)$
3. $\forall\, u \in (\Sigma \times \boldsymbol{T})^+,\ u \in \mathscr{G}(\alpha) \Leftrightarrow \Lambda(\alpha, u) \neq \varnothing$

Rappelons que $\wp(E)$ est l'ensemble des parties de l'ensemble E

A partir des résultats précédents, il est clair que pour tout automate temporisé $\mathscr{A}$, le couple $(\mathrm{L}_{\mathscr{A}}, \mathrm{Ends}_{\mathscr{A}})$ est un générateur contraint associé à $\mathscr{A}$.

Deux automates temporisés $\mathscr{A}$ et $\mathscr{B}$ sont dits équivalents, noté $\mathscr{A} = \mathscr{B}$, si leurs générateurs contraints associés $\hat{\mathscr{A}}$ et $\hat{\mathscr{B}}$ sont égaux. Dans ce cas, on a en particulier $\mathrm{L}_{\mathscr{A}}(\mathbf{0}) = \mathrm{L}_{\mathscr{B}}(\mathbf{0})$.

3. Composition d'automates temporisés

3.1. Union

L'union de deux automates temporisés $\mathscr{A} = (Q_1, T_1, I_1, F_1, R_1)$ et $\mathscr{B} = (Q_2, T_2, I_2, F_2, R_2)$ avec $Q_1 \cap Q_2 = \varnothing$ est simplement l'automate temporisé $\mathscr{A} + \mathscr{B} = (Q_1 \cup Q_2, T_1 \cup T_2, I_1 \cup I_2, F_1 \cup F_2, R_1 \cup R_2)$.

3.2. Composition

Le principal problème dans la composition de systèmes temporisés vient des horloges. Supposons que nous voulions concaténer deux automates temporisés $\mathscr{A}$ et $\mathscr{B}$. Alors, lorsque l'automate résultant de la concaténation effectue pour la première fois une transition vers un état de $\mathscr{B}$, il y deux possibilités, soit réinitialiser toutes les horloges de $\mathscr{A}$ soit au contraire ne réinitialiser aucune horloge de $\mathscr{A}$. Mais dans de nombreux cas, ces possibilités extrêmes ne sont pas satisfaisantes. Supposons par exemple que nous voulions construire un produit complexe ayant des contraintes de temps globales et que cette construction se fasse en plusieurs parties, chacune ayant ses propres contraintes de temps. Il est alors plus naturel de permettre, dans la concaténation des automates temporisés modélisant chaque sous-système, de réinitialiser certaines horloges et pas d'autres.

En suivant cette idée, simple mais puissante, la concaténation de deux automates temporisés est définie en fonction d'un ensemble fixe d'horloges. Supposons que deux automates temporisés $\mathscr{A}$ et $\mathscr{B}$ et un ensemble d'horloges C soient donnés (avec $x_0 \notin C$). Intuitivement, la concaténation $\mathscr{A}_C\mathscr{B}$ est donnée par la Figure 1 (avec y une horloge utilisée ni dans $\mathscr{A}$ ni dans $\mathscr{B}$).

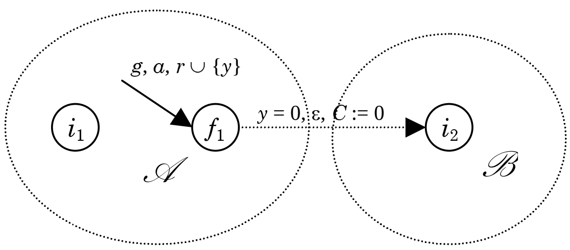

Figure 1 : Automate $\mathscr{A}_C\mathscr{B}$

L'utilisation d'ε–transitions n'est pas permise dans notre modèle. Néanmoins, il est facile de vérifier que $\mathscr{A} \dot{_C} \mathscr{B}$ peut-être défini formellement, sans utiliser de telles transitions, de la manière suivante. Soit $\mathscr{A} = (Q_1, T_1, I_1, F_1, R_1)$ et $\mathscr{B} = (Q_2, T_2, I_2, F_2, R_2)$ avec $Q_1 \cap Q_2 = \varnothing$. Nous considérons une copie $\widetilde{F}_1$ de F_1, disjoint de $Q_1 \cup Q_2$. Alors $\mathscr{A} \dot{_C} \mathscr{B}$ est l'automate temporisé $(Q_1 \cup Q_2 \cup \widetilde{F}_1, \mathrm{T}, I_1, F_2, R_1 \cup R_2)$

avec

$$\mathrm{T} = T_1 \cup T_2 \cup \{(q_1, g_1, a_1, r_1 \cup C, \tilde{f}_1) \mid f_1 \in F_1, (q_1, g_1, a_1, r_1, f_1) \in T_1\}$$

$$\cup \{(\tilde{f}_1, g_2, a_2, r_2, q_2) \mid \exists\, i_2 \in I_2, (i_2, g_2, a_2, r_2, q_2) \in T_2\}$$

3.3. Itérations

Nous dérivons maintenant de l'opérateur de concaténation $\dot{_C}$ les deux opérateurs $^{\dot{+}}_{C}$ et $^{\omega}_{C}$ qui correspondent aux itérations finie et infinie. Soit $A = (Q, T, I, F, R)$ un automate temporisé et soit C un sous-ensemble d'horloges. Nous considérons une copie $\widetilde{F}$ de F, disjoint de Q. Alors l'automate temporisé $A^{\dot{+}_C}$ est défini comme suit, $A^{\dot{+}_C} = (Q \cup \widetilde{F}, T', I, \widetilde{F}, R)$ avec

$$T' = \mathrm{T} \cup \{(q, g, a, r \cup C, \tilde{f}) \mid f \in F, (q, g, a, r, f) \in T\}$$

$$\cup \{(\tilde{f}, g, a, r, q) \mid \exists\, i \in I, (i, g, a, r, q) \in T\}$$

De façon similaire, l'itération infinie de A, notée A^{ω_C}, est définie par $A^{\omega_C} = (Q \cup \widetilde{F}, T', I, \varnothing, R \cup \widetilde{F})$ où T' est défini ci-dessus.

3.4. Construction modulaire d'automates temporisés

Pour toute garde g ∈ Guards(X) et toute action $a \in \Sigma$, nous notons $A_{<\delta\, a>g}$ l'automate simple temporisé de la Figure 2. Pour toute valuation d'horloge $\alpha \in \boldsymbol{T}^X$ et tout $\mathrm{t} \in \boldsymbol{T}$, nous notons $\alpha + \mathrm{t}$ la valuation d'horloge définie pour toute horloge x par, $(\alpha + \mathrm{t})(x) = \alpha(x) + \mathrm{t}$. Il devient alors immédiat de vérifier que $\mathrm{L}_{\mathscr{A}<\delta\, a>\mathrm{g}}(\alpha)$ est égal à $\{ (a, \mathrm{t}) \mid \alpha + \mathrm{t} \models \mathrm{g} \}$.

De plus, $\mathrm{Ends}_{A<\delta\, a>\mathrm{g}}(\alpha, a) = \{ \alpha + \mathrm{t} \mid \alpha + \mathrm{t} \models \mathrm{g} \}$.

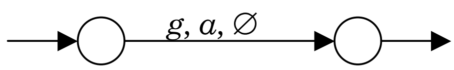

Figure 2 : Automate $A_{<\delta\, a>g}$

A partir des automates de base et des opérateurs de composition définis ci-dessus, nous définissons maintenant une famille modulaire d’automates temporisés.

Définition 2 : Soit Mod(Σ, $\boldsymbol{T}$, X) le plus petit ensemble d’automates temporisés de TA(Σ, $\boldsymbol{T}$, X), les éléments de cet ensemble sont les automates de la forme :

- $A_{<\delta\, a>g}$ où a est une action de Σ, $g \in$ Guards(X) une garde sur X et $C \subseteq X$ un sous-ensemble d’horloges tel que $x_0 \notin C$.
- $A_1 + A_2$, $A_1 \dot{}_C A_2$, $A_1^{\overset{+}{C}}$ et $A_1^{\overset{\varpi}{C}}$ ssi A_1 et A_2 appartiennent à Mod(Σ, $\boldsymbol{T}$, X)

Nous allons prouver le théorème 3 en résolvant des systèmes d’équations sur les générateurs contraints. N’étant pas sûr, a priori, que la solution est toujours un générateur contraint associé à un automate temporisé, il nous faut définir certaines opérations sur les générateurs contraints.

Théorème 3 : tout automate temporisé de TA(Σ, $\boldsymbol{T}$, X) est équivalent à un automate de Mod(Σ, $\boldsymbol{T}$, X).

4. Composition de générateurs contraints

Le but de cette section est d'obtenir des définitions telles que la proposition suivante se tienne.

Proposition 4 : Soit A et B deux automates temporisés de TA(Σ, $\boldsymbol{T}$, X) et $C \subseteq X$ un sous-ensemble d’horloges. Alors, on a :

1. $\widehat{A + B} = \widehat{A} + \widehat{B}$
2. $\widehat{A \dot{}_C B} = \widehat{A} \dot{}_C \widehat{B}$
3. $\widehat{\mathrm{A}^{\overset{+}{C}}} = \widehat{\mathrm{A}}^{\overset{+}{C}}$
4. $\widehat{A^{\overset{\varpi}{C}}} = \widehat{A}^{\overset{\varpi}{C}}$

4.1 Union de générateurs contraints

Dans cette section, ($\mathscr{G}$, Λ) et ($\mathscr{G}$’, Λ’) sont deux générateurs contraints et $C \subseteq X$ est un sous-ensemble d'horloges.

L’union de ($\mathscr{G}$, Λ) et ($\mathscr{G}$’, Λ’), noté ($\mathscr{G}$, Λ) + ($\mathscr{G}$’, Λ’), est naturellement le générateur contraint ($\mathscr{G}$’’, Λ’’) où, pour toute valuation d’horloge $\alpha \in \boldsymbol{T}^X$ et tout mot temporisé fini $w \in (\Sigma \times \boldsymbol{T})^+$, $\mathscr{G}$’’(α) = $\mathscr{G}$(α) $\cup$ $\mathscr{G}$’(α) et Λ’’(α, u) = Λ(α, u) $\cup$ Λ’(α, u).

4.2 Composition de générateurs contraints

Nous définissons la composition $(\mathscr{G}, \Lambda)\ \dot{c}\ (\mathscr{G}', \Lambda')$ comme le générateur contraint $(\mathscr{G}'', \Lambda'')$ où, pour toute valuation d'horloge $\alpha \in \boldsymbol{T}^X$ et tout mot temporisé[1] fini $w \in (\Sigma \times \boldsymbol{T})^+$,

$$\mathscr{G}''(\alpha) = \{\, u.v \mid u \in \mathscr{G}(\alpha) \cap (\Sigma \times \boldsymbol{T})^+ \text{ et } v \in \mathscr{G}'(\beta[C := 0]) \cap (\Sigma \times \boldsymbol{T})^\infty \text{ pour } \beta \in \Lambda(\alpha, u) \,\}$$

et $\Lambda''(\alpha, w) = \{\, \Lambda'(\beta[C := 0], v) \mid \text{il existe } u \in \mathscr{G}(\alpha) \text{ avec } w = u.v \text{ et } \beta \in \Lambda(\alpha, u) \,\}$

4.3 Itérations de générateurs contraints

Nous définissons maintenant de façon inductive, le générateur contraint , $(\mathscr{G}, \Lambda)^{(\dot{c})^i}$, pour $i \geq 1$, par $(\mathscr{G}, \Lambda)^{(\dot{c})^1} = (\mathscr{G}, \Lambda)$ et $(\mathscr{G}, \Lambda)^{(\dot{c})^{i+1}} = (\mathscr{G}, \Lambda)\ \dot{c}\ (\mathscr{G}, \Lambda)^{(\dot{c})^i}$

Et nous posons $(\mathscr{G}, \Lambda)^{\overset{+}{c}} = \Sigma_{i \geq 1}\, (\mathscr{G}, \Lambda)^{(\dot{c})^i}$.

Enfin, soit $(\mathscr{G}, \Lambda)$ un générateur contraint et $(\mathscr{G}_1, \Lambda_1) = (\mathscr{G}, \Lambda)^{\overset{+}{c}}$. Nous définissons $(\mathscr{G}, \Lambda)^{\overset{\omega}{c}} = (\mathscr{G}'', \Lambda'')$ par : pour toute valuation d'horloge $\alpha \in \boldsymbol{T}^X$ et tout mot temporisé fini $w \in (\Sigma \times \boldsymbol{T})^+$,

$\mathscr{G}''(\alpha) = (\mathscr{G}_1(\alpha) \cap (\Sigma \times \boldsymbol{T})^\omega) \cup \{\, u_0.u_1\ldots \mid$ il existe une séquence $(\alpha_i)_{i \geq 0}$ de valuations d'horloges avec $\alpha_0 = \alpha$ pour tout $i \geq 1$, $u_{i+1} \in \mathscr{G}(\alpha_i) \cap (\Sigma \times \boldsymbol{T})^+$ et $\alpha_{i+1} \in \Lambda(\alpha_i, u_i)\,\}$

Et $\Lambda''(\alpha, w) = \emptyset$.

Avec ces définitions, la preuve de la proposition 4 est maintenant technique mais sans difficulté majeure.

4.4 Propriétés de base

Le résultat suivant récapitule les propriétés des opérateurs $\dot{c}$ et $\overset{+}{c}$ nécessaires pour la suite.

Proposition 5 : Soient $(\mathscr{G}, \Lambda)$, $(\mathscr{G}', \Lambda')$ et $(\mathscr{G}'', \Lambda'')$ trois générateurs contraints et soient C, D deux sous-ensembles d'horloges. Alors, on a :

1. $(\mathscr{G}, \Lambda)\ \dot{c}\ ((\mathscr{G}', \Lambda') + (\mathscr{G}'', \Lambda'')) = ((\mathscr{G}, \Lambda)\ \dot{c}\ (\mathscr{G}', \Lambda')) + ((\mathscr{G}, \Lambda)\ \dot{c}\ (\mathscr{G}'', \Lambda''))$

2. $((\mathscr{G}, \Lambda) + (\mathscr{G}', \Lambda')) \ \dot{c}\ (\mathscr{G}'', \Lambda'') = ((\mathscr{G}, \Lambda) \ \dot{c}\ (\mathscr{G}'', \Lambda'')) + ((\mathscr{G}', \Lambda') \ \dot{c}\ (\mathscr{G}'', \Lambda''))$

3. $((\mathscr{G}, \Lambda) \ \dot{c}\ (\mathscr{G}', \Lambda')) \ \dot{D}\ (\mathscr{G}'', \Lambda'') = (\mathscr{G}, \Lambda) \ \dot{c}\ ((\mathscr{G}', \Lambda') \ \dot{D}\ (\mathscr{G}'', \Lambda''))$

4. $(\mathscr{G}, \Lambda)^{\overset{+}{c}} \ \dot{c}\ (\mathscr{G}', \Lambda') = \Sigma_{i\geq 1}(\mathscr{G}, \Lambda)^{(\cdot)} \ \dot{c}\ (\mathscr{G}', \Lambda')$

4.5 Equations sur les générateurs contraints

Nous considérons deux générateurs contraints $(\mathscr{G}_1, \Lambda_1)$ et $(\mathscr{G}_2, \Lambda_2)$ et un sous-ensemble C d'horloges. Le lemme suivant est le résultat fondamental permettant de résoudre le système d'équations sur les générateurs contraints.

Lemme 6 : L'équation sur les générateurs contraints

$$(\mathscr{G}, \Lambda) = (\mathscr{G}_1, \Lambda_1) \ \dot{c}\ (\mathscr{G}, \Lambda) + (\mathscr{G}_2, \Lambda_2)$$

a pour solution unique le générateur contraint $(\mathscr{G}_1, \Lambda_1)^{\overset{+}{c}} \ \dot{c}\ (\mathscr{G}_2, \Lambda_2) + (\mathscr{G}_2, \Lambda_2)$.

5. Décomposition d'automates temporisés

Nous sommes maintenant prêts à prouver notre résultat principal, le théorème 3. Soit $A = (Q, T, I, F, R)$ un automate temporisé. Nous supposons que pour n'importe quel état q il existe des sous-ensembles $C_q \subseteq X$ tels que pour n'importe quelle transition (q', g, a, r, q), on a $r = C_q$. Notez que, en remplaçant l'ensemble des états Q par le produit cartésien $Q \times \wp(X)$, il est facile de transformer n'importe quel automate temporisé en automate temporisé équivalent vérifiant cette propriété. Nous proposerons maintenant un algorithme pour trouver un automate temporisé B dans Mod(Σ, $\boldsymbol{T}$, X) qui soit équivalent à A.

Pour tous les états i, f $\in Q$, nous prenons $A_{i,f} = (Q, T; \{i\}; \{f\}, \varnothing)$ et nous considérons le générateur contraint $\widehat{A_{i,f}}$ associé à $A_{i,f}$. A partir de la définition de l'exécution d'un automate temporisé (voir la section 2), il est facile de vérifier l'équation suivante

$$\hat{A} = \sum_{i\in I, f\in F} \widehat{A_{i,f}} + \sum_{i\in I, q\in R} \widehat{A_{i,q}} \ \dot{c}_q\ \widehat{A_{q,q}}^{\overset{\wp}{\mathscr{E}}_q}$$

ce qui, en terme d'automates temporisés et en utilisant la proposition 4, peut être réécrit de manière équivalente en

[1] Rappelons que la concaténation de deux mots temporisés est une opération partielle définie dans la section 2.1

$$A \equiv \sum_{i\in I, f\in F} A_{i,f} + \sum_{i\in I, f\in F} A_{i,q} \, \dot{c}_q \, A_{q,q}^{\overset{\omega}{c}_q} \qquad (\Psi)$$

Par conséquent, afin de montrer que A est équivalent à un automate de Mod(Σ, $\boldsymbol{T}$, X), il suffit de prouver que pour tout q, q' $\in$ Q, l'automate $A_{q,\,q'}$ appartient à Mod(Σ, $\boldsymbol{T}$, X).

Supposons maintenant que f $\in$ Q soit fixé, puis que la famille des générateurs contraints $(\widehat{A_{q,f}})_{q \in Q}$ vérifie le système d'équations sur les générateurs contraints - où $\langle \delta\, a \rangle_g$ dénote le générateur contraint associé à l'automate temporisé de base $A_{\langle \delta\, a \rangle\, g}$ défini dans la section 3.4 :

$$\begin{cases} \widehat{A_{q,f}} = \varepsilon + \sum_{(q,g,a,C_{q'},q')\in T} \langle \delta a \rangle_g \dot{r} \ \widehat{A_{q',f}} & \text{si } q = f \\ \widehat{A_{q,f}} = \sum_{(q,g,a,C_{q'},q')\in T} \langle \delta a \rangle_g \dot{r} \ \widehat{A_{q',f}} & \text{sinon} \end{cases}$$

Considérons maintenant un ordre arbitraire $q_1 < q_2 < \ldots < q_n$ sur les éléments de Q. L'équation dont le membre gauche est $\widehat{A_{q_n,f}}$ est alors résolue, avec $\widehat{A_{q_n,f}}$ en tant qu'inconnue, en utilisant le lemme fondamental 6 - ε doit être ajouté si $q_n = f$:

$$\widehat{A_{q_n,f}} = \left(\sum_{(q_n,g,a,C_{q_n},q_n)\in T} \langle \delta a \rangle_g \right)^{\overset{+}{c}_{q_n}} \dot{c}_{q_n} \left(\varepsilon_{q,f} + \sum_{(q_n,g,a,C_{q'},q')\in T,\, q'\neq q_n} \hat{A}_{q',f} \right)$$

Nous remplaçons alors $\widehat{A_{q_n,f}}$ par cette formule dans les n - 1 autres équations. Pas à pas, nous résolvons le système en utilisant le lemme 6. La dernière étape prouve que le générateur contraint $\widehat{A_{q_1,f}}$ peut être exprimé en utilisant des générateurs contraints élémentaires du type $\langle \delta\, a \rangle_g$, pour tout a $\in$ Σ et les opérateurs de composition, ce qui revient à dire que l'automate $A_{q_1,f}$ est équivalent à un automate de Mod(Σ, $\boldsymbol{T}$, X). Nous en déduisons que $A_{q_2,f}$, puis $A_{q_3,f}$, …, $A_{q_n,f}$ sont également équivalents à un automate de Mod(Σ, $\boldsymbol{T}$, X).

Finalement, pour tout q, q' $\in$ Q, tout automate $A_{q,q'}$ est équivalent à un automate de Mod(Σ, $\boldsymbol{T}$, X), et donc en utilisant l'équation (Ψ), A est également équivalent à un automate de Mod(Σ, $\boldsymbol{T}$, X). Le théorème 3 est donc prouvé.

Cette méthode est illustrée dans un exemple de l'annexe A.

Annexe A : exemple

Nous illustrons l'algorithme sur l'exemple d'automate suivant.

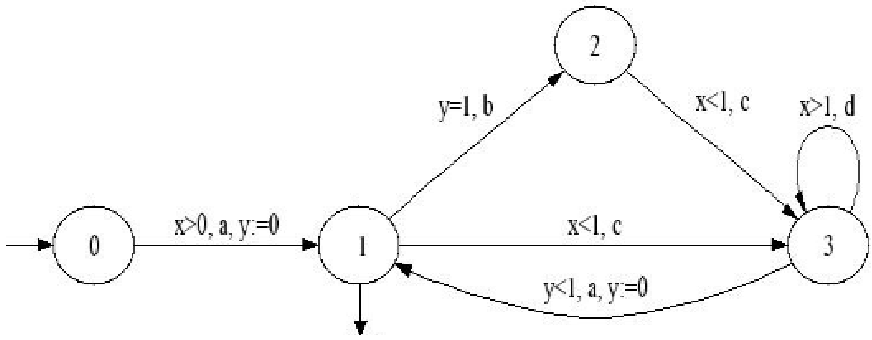

Figure 3 : Automate A

L'automate A de la Figure 3 définit le système d'équations linéaires temporisées suivant :

$$\begin{cases} X_0 = \langle \delta a \rangle_{x>0,a,y:=0} X_1 \\ X_1 = \varepsilon + \langle \delta b \rangle_{y=1,b} X_2 + \langle \delta c \rangle_{x<1,c} X_3 \\ X_2 = \langle \delta c \rangle_{x<1,c} X_3 \\ X_3 = \langle \delta d \rangle_{x>1,d} X_3 + \langle \delta a \rangle_{y<1,a,y:=0} X_1 \end{cases}$$

En utilisant l'algorithme présenté dans la section 5, on peut résoudre le système d'équations et obtenir comme solution :

$$X_0 = \langle \delta a \rangle_{x>0,a,y:=0} \left(\varepsilon + \left(\langle \delta b \rangle_{y=1,b} \langle \delta c \rangle_{x<1,c} Z + \langle \delta c \rangle_{x<1,c} Z \right)^{+}_{\{y\}} \right)$$

où

$$Z = \left(\varepsilon + \left(\langle \delta d \rangle_{x>1,d} \right)^{+}_{\emptyset} \right) \dot{\emptyset} \langle \delta a \rangle_{y<1,a,y:=0}$$

Ceci définit donc une décomposition modulaire de A en automates temporisés simples comme $A\langle \delta a \rangle_{x>0,a}$, $A\langle \delta c \rangle_{x<1,c}$, …