Premiers pas vers le diagnostic incrémental de systèmes à événements discrets

Alban Grastien[1], Marie-Odile Cordier[1], and Christine Largouët[2]

[1] Irisa, Université de Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex
{agrastie,cordier}@irisa.fr
[2] University of New Caledonia, BP. 4477, 98847 Nouméa Cedex, New Caledonia,
largouet@univ-nc.nc

Résumé Nous nous intéressons au calcul incrémental du diagnostic hors-ligne d'un système modélisé comme un système à événements discrets par un automate. Étant donné un automate représentant les observations émises par le système sur une période donnée, nous voulons, au lieu de calculer globalement le diagnostic pour cet automate des observations, découper les observations selon des fenêtres temporelles, calculer les diagnostics locaux expliquant chacune de ces périodes d'observations, et montrer qu'il est possible de retrouver le diagnostic global à partir de ces diagnostics locaux. Pour atteindre cet objectif, nous proposons le concept de chaîne d'automates. Après avoir proposé une formalisation de ce calcul par morceaux du diagnostic, nous montrons que l'on peut profiter du diagnostic local obtenu pour la fenêtre précédente pour améliorer le calcul du diagnostic de la fenêtre courante. Il s'agit bien alors de diagnostic incrémental. Cette étude est la première étape nécessaire d'un travail plus ambitieux dont l'objectif est le diagnostic incrémental dans le cadre de la surveillance en-ligne. Les deux principales difficultés sont alors le découpage correct et en-ligne de l'automate des observations, et la détermination des fenêtres temporelles. Dans cet article, nous supposons que le découpage est effectué et vérifie les propriétés, que nous énonçons et qui sont nécessaires au calcul incrémental du diagnostic.

1 Introduction

Dans le domaine du diagnostic des systèmes à événements discrets, il est assez classique de définir le diagnostic comme l'ensemble des trajectoires compatibles avec les observations. Différentes terminologies peuvent être retrouvées dans la littérature, comme *histories* [1], *scenarios* [2], *narratives* [3], *consistent paths* [4] ou *trajectories* [5]. On caractérise alors le calcul du diagnostic lui-même comme résultant de la synchronisation de l'automate modélisant le comportement du système avec l'automate décrivant les observations émises par le système sur toute la période concernée. Il est possible de résoudre le calcul plus simplement en considérant que les observations sont reçues sans perte ni délai par le superviseur en charge du diagnostic, mais cette hypothèse n'est pas valable pour des systèmes réels (voir par exemple [6]). Cette caractérisation formelle suppose que l'on dispose de l'ensemble des observations. Nous nous plaçons pour cet article dans le cadre hors-ligne. Ainsi cette hypothèse est vérifiée. Cependant, la taille de cet automate des observations peut être trop importante et dépend directement de la durée des observations. Par exemple, on peut penser à un diagnostic *a posteriori* à partir d'observations collectées sur une période de plusieurs jours, comme c'est le cas pour des logs d'alarmes dans le domaine des réseaux de télécommunications. Dans cet article nous cherchons à savoir s'il est possible d'éviter un calcul global en découpant l'ensemble d'observations selon des fenêtres temporelles (période de temps de durée variable).

Pour cela, nous proposons le concept de chaîne d'automates qui permet de représenter observations et diagnostics de manière modulaire (par morceaux) tout en garantissant la reconstruction de l'automate global à partir de la chaîne d'automates résultant du découpage.

Nous montrons qu'il est possible de calculer pour chaque fenêtre temporelle le diagnostic local correspondant et que le diagnostic global est alors bien représenté par la chaîne d'automates de diagnostics locaux. Ainsi, nous proposons une première formalisation de ce calcul par morceaux du diagnostic. Un point noir est cependant la taille éventuellement importante de ces diagnostics locaux qui sont construits sans tenir compte du contexte de la fenêtre temporelle, en particulier sans tenir compte des diagnostics faits lors des fenêtres précédentes. Nous montrons alors comment il est possible de profiter du diagnostic local de la fenêtre précédente pour améliorer le calcul du diagnostic de la fenêtre courante. Il s'agit alors bien de diagnostic incrémental. Ce travail est la première étape nécessaire d'un travail plus ambitieux dont l'objectif est le diagnostic incrémental dans le cadre de la surveillance en-ligne. Dans cet article, nous supposons que le découpage est effectué et vérifie les propriétés, que nous énonçons et qui sont nécessaires au calcul incrémental du diagnostic.

Le formalisme employé pour représenter le comportement du système (modèle), les observations et le diagnostic est défini dans la section 2. La problématique ainsi que les différentes étapes permettant l'élaboration du diagnostic incrémental sont présentées dans la section 3. La section 4 décrit le concept de *chaîne d'automates* qui est central pour la formalisation et le calcul incrémental du diagnostic que nous proposons. Les deux versions de ce calcul sont ensuite détaillées dans la section 5.

2 Diagnostic global

Cette section aborde le diagnostic calculé de manière globale sans considérer les problèmes posés par l'incrémentalité. Les définitions utilisées pour le calcul du diagnostic global constituent également le fondement théorique pour le calcul incrémental que nous abordons par la suite.

2.1 Automates et trajectoires

Le système réagit à la suite de l'occurence d'événements. Certains événements peuvent être simultanés, notamment si l'on considère que l'occurence de certains événements peut déclencher d'autres événements (cas des systèmes réactifs, voir par exemple [7]). On note E, l'ensemble des événements. Nous représentons les comportements par des automates, définis de manière traditionnelle :

Définition 1 (Automate) *On appelle* automate *le t-uplet* $A = (Q, E, T, I, F)$ *où :*
- Q *est un ensemble d'états,*
- E *est un ensemble d'événements,*
- $T \subseteq (Q \times L(E) \times Q)$ *est un ensemble de transitions* $t = (q, l, q')$ *où* q *est l'état de départ,* q' *l'état d'arrivée et* l *un ensemble non vide d'événements* ($l \subseteq E$)*,*
- I *est un ensemble d'états initiaux* ($I \subseteq Q$)*,*
- F *est un ensemble d'états finaux* ($F \subseteq Q$)*.*

Les étiquettes de transition des automates ne doivent pas être vides. On considère cependant implicitement que $\forall q \in Q$, la transition $(q, \emptyset, q)$ est une transition de T.

Définition 2 (Trajectoire) *On appelle* trajectoire *notée* $traj$ *sur un automate* $A = (Q, E, T, I, F)$ *la double séquence* finie *d'états* $(q_0, \ldots, q_n)$ *et d'étiquettes* $(l_1, \ldots, l_n)$ *telle que :*
- $\forall i \in \{0, \ldots, n\},\ q_i \in Q$*,*
- $\forall i \in \{1, \ldots, n\},\ t_i = (q_{i-1}, l_i, q_i) \in T$*,*
- $q_0 \in I$ *et*
- $q_n \in F$*.*

Une trajectoire est décrite par une séquence d'états dont le premier est un état initial et le dernier un état final, et une séquence d'étiquettes sur les transitions permettant de passer d'un état à l'état suivant.

Une trajectoire peut comporter des transitions implicites sur l'automate. Elle est dite identique à une trajectoire dont on a supprimé les transitions implicites. Plus formellement, soit $traj = ((q_0, \ldots, q_i, q_{i+1}, \ldots, q_n), (l_0, \ldots, l_i, l_{i+1}, \ldots, l_n))$ et $traj' = ((q_0, \ldots, q_i, q_i, q_{i+1} \ldots, q_n), (l_0, \ldots, l_i, \emptyset, l_{i+1}, \ldots, l_n))$. Alors $traj = traj'$.

Deux automates A et A' sont dits identiques ($A = A'$) si leurs ensembles de trajectoires sont identiques. On appelle *automate simplifié* de A, l'automate $A' = A$ dans lequel tous les états et toutes les transitions non accessibles depuis un état initial ou ne permettant pas d'atteindre un état final ont été supprimés. Par la suite, lorsque nous calculons de nouveaux automates, nous considérons de manière implicite que nous en calculons l'automate simplifié.

2.2 Automates synchronisés

Définition 3 (Synchronisation d'étiquettes) *Soit* l_1 *une étiquette sur* E_1 *et* l_2 *une étiquette sur* E_2*. On dit que* l_1 *et* l_2 *sont* synchrones *ssi* $l_1 \cap (E_1 \cap E_2) = l_2 \cap (E_1 \cap E_2)$*. Leur synchronisation notée* $\Theta(l_1, l_2)$ *est l'étiquette* $l_1 \cup l_2$ *sur l'ensemble d'événements* $E_1 \cup E_2$*.*

Deux étiquettes sont synchrones si les événements de synchronisation ($E_1 \cap E_2$) présents dans l'une des étiquettes sont également présents dans l'autre.

Définition 4 (Synchronisation) *Soient* $A_1 = (Q_1, E_1, T_1, I_1, F_1)$ *et* $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ *deux automates. On appelle* automate synchronisé *de* A_1 *et* A_2*, noté* $A_1 \otimes A_2$*, l'automate* $A = (Q, E, T, I, F)$ *défini par :*
- $Q = Q_1 \times Q_2$,
- $E = E_1 \cup E_2$,
- $T = \{((q_1, q_2), l, (q_1', q_2')) \mid \exists l_1, l_2,$
  - $(q_1 = q_1' \wedge l_1 = \emptyset) \vee (q_1, l_1, q_1') \in T_1$
  - $(q_2 = q_2' \wedge l_2 = \emptyset) \vee (q_2, l_2, q_2') \in T_2$
  - $l = \Theta(l_1, l_2)$

  $\}$,
- $I = I_1 \times I_2$,
- $F = F_1 \times F_2$.

La synchronisation consiste à emprunter deux transitions dont les étiquettes sont synchrones simultanément sur les deux automates.

On peut facilement prouver que $(A_1 \otimes A_2) \otimes A_3 = A_1 \otimes (A_2 \otimes A_3)$ avec le renommage des états suivant : $((q_1, q_2), q_3) \rightarrow (q_1, (q_2, q_3))$. Par la suite, et pour simplifier, nous notons donc : $A = A_1 \otimes \ldots \otimes A_n = A_1 \otimes (\ldots \otimes A_n) = (A_1 \otimes \ldots) \otimes A_n = (Q, E, T, I, F)$ avec $Q = Q_1 \times \ldots \times Q_n$. De même, on ne fera pas de différence entre $A_1 \otimes A_2$ et $A_2 \otimes A_1$. De manière plus générale, on ne fera pas de différence entre un état $((q_1, q_2), q_3)$ et un état (q_1, q_2, q_3).

2.3 Diagnostic

Définition 5 (Modèle du système) *Le* modèle du système*, noté MOD est un automate* $(Q^{MOD}, E^{MOD}, T^{MOD}, I^{MOD}, F^{MOD})$. I^{MOD} *représente l'ensemble des états possibles à* t_0*. Tous les états du modèle sont potentiellement finaux (*$F^{MOD} = Q^{MOD}$*). L'ensemble des événements observables est* $E_{OBS}^{MOD} \subseteq E^{MOD}$.

Considérons les observations. De manière générale, on ne dispose pas d'ordre total sur les observations émises par le système. Par conséquent, les observations sont représentées par un automate, dont chaque trajectoire représente un ordre possible d'émission des observations durant la période $[t_0, t_n]$.

Définition 6 (Observations) *Les* observations*, notées* OBS_n *sont un automate décrivant les observations émises par le système durant la période* $[t_0, t_n]$.

Définition 7 (Diagnostic) *Le* diagnostic*, noté* Δ_n *est un automate décrivant les trajectoires possibles sur le modèle du système compatibles avec les observations émises par le système durant la période* $[t_0, t_n]$.

Le diagnostic global du système peut être calculé de la manière suivante (voir par exemple [8,9]) :

$$\Delta_n = OBS_n \otimes MOD \tag{1}$$

3 Problématique

Nous avons vu que le diagnostic global Δ_n se construit à partir de l'automate représentant les observations durant la totalité de la fenêtre temporelle. Nous cherchons à présent à calculer par étapes le diagnostic du système sur une période $[t_0, t_n]$, avec t_n non fixé. À partir du diagnostic de la période $[t_0, t_i]$ et de nouvelles informations sur les observations portant jusqu'à la période t_{i+1}, nous souhaitons calculer le diagnostic de la période $[t_0, t_{i+1}]$.

Dans un premier temps, nous divisons le temps en périodes appelées *fenêtres temporelles*. La fenêtre temporelle $\mathcal{W}^i$ correspond à l'intervalle de temps délimité par $[t_{i-1}, t_i]$. À chaque fenêtre temporelle correspond un ensemble d'observations émises par le système. C'est la raison pour laquelle nous *découpons* l'automate des observations en une *chaîne d'automates*, chaque automate de la chaîne correspondant à une fenêtre temporelle. Chaque automate nous permet de calculer alors le diagnostic Δ^i *local* à la fenêtre temporelle $\mathcal{W}^i$. Enfin, le *diagnostic partiel* du système correspondant à la période $[t_0, t_i]$ et noté Δ_i peut être calculé. Le diagnostic partiel résume toutes les informations concernant la tranche de temps $[t_0, t_i]$ alors que le diagnostic local ne s'intéresse qu'à la période $[t_{i-1}, t_i]$.

Dans un deuxième temps, nous définissons le calcul incrémental du diagnostic. On dispose pour cela de Δ_i, le diagnostic de la période $[t_0, t_i]$. Il faut calculer le diagnostic Δ_{i+1} de la période $[t_0, t_{i+1}]$ à partir de Δ_i et du diagnostic local Δ^{i+1}.

4 Chaînes d'automates

Dans cette section, nous introduisons une représentation particulière appelée *chaîne d'automates*. Cette représentation nous permet de calculer le diagnostic par fenêtres temporelles.

Le principe est présenté sur la figure 1. Partant de l'automate des observations OBS_n et du modèle MOD du système, il est possible, comme présenté en section 2, de construire le diagnostic global du système par synchronisation avec le modèle. Nous supposons que l'automate des observations est *découpé* en une chaîne d'automates locaux. Ce découpage est considéré comme correct s'il est possible de retrouver l'automate des observations original par une opération appelée *reconstruction*. À partir de ces automates des observations locaux OBS^i à une fenêtre temporelle $\mathcal{W}^i$, nous allons calculer les diagnostics locaux Δ^i. Le but est de calculer le diagnostic global du système à partir de cette chaîne de diagnostic locaux.

Nous présentons les chaînes d'automates et les liens entre la chaîne d'automates et l'automate qu'elle représente. Après quoi, nous donnons les propriétés des chaînes d'automates qui nous permettent de définir le diagnostic.

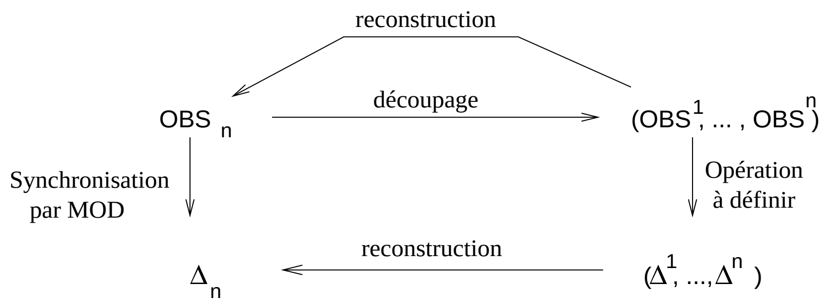

Fig. 1. Principe de la chaîne d'automates

4.1 $\Upsilon-$transition

On distingue un type d'événement particulier : les $\Upsilon-$événements (notés Υ_i pour $i \in \mathbf{N}$). Un $\Upsilon-$événement correspond à un top d'horloge. Il n'a pas d'autre sémantique. En particulier, ce n'est pas un événement sur le système.

Nous considérons qu'aucun événement ne peut avoir lieu en même temps qu'un Υ-événement. Nous ajoutons à la définition des étiquettes l la propriété suivante : si l est tel que $\exists k, \Upsilon_k \in l$, alors $l = \{\Upsilon_k\}$ Par la suite, nous écrivons $l = \Upsilon_k$ pour $l = \{\Upsilon_k\}$. On appelle $\Upsilon-$transition une transition étiquetée par Υ_k.

Nous modifions également la définition de la synchronisation en ajoutant la propriété suivante : pour être synchronisées, deux étiquettes l_1 et l_2 doivent être telles que si $(\exists i \in \{1,2\}, j \in \{1,2\}, j \neq i),\ l_i = \Upsilon_k$, alors $l_j = l_i$ ou $l_j = \emptyset$. Cette propriété permet de s'assurer qu'une étiquette obtenue par synchronisation d'étiquettes respecte la propriété des $\Upsilon-$événements.

Les tops d'horloge représentés par les Υ-transitions permettent de définir des fenêtres temporelles. On appelle *fenêtre temporelle*, notée $\mathcal{W}^i$, la période délimitée par le top Υ_{i-1} et le top Υ_i. La période $\mathcal{W}^1$ est la période précédant le premier top Υ_1 et la période $\mathcal{W}^n$ est la période suivant Υ_{n-1} (où Υ_{n-1} est le dernier top d'horloge).

4.2 Chaînes d'automates

Définition 8 (Chaîne d'automates) *Une séquence d'automates* $(A^1, \ldots, A^n)$ *avec* $A^i = (Q^i, E^i, T^i, I^i, F^i)$ *est appelée* chaîne d'automates*, et notée* $\mathcal{E}_A$*, si elle respecte la propriété suivante :* $\forall i \in \{1, \ldots, n\}, \forall k \in \mathbf{N},\ \Upsilon_k \notin E^i$.

La sémantique d'une chaîne d'automates est la suivante : on se déplace tout d'abord sur l'automate A^1 à partir d'un état de I^1. Puis, une fois arrivés dans un état q de $F^1 \cap I^2$, on passe sur l'automate A^2 et on se déplace sur cet automate A^2 à partir de ce même état q, *etc.* L'automate A^i est associé à la fenêtre temporelle $\mathcal{W}^i$, *etc.* Une chaîne $(A^1, \ldots, A^n)$ pourra aussi être représentée de la manière suivante : $((A^1, \ldots, A^{n-1}), A^n)$.

Si la chaîne comporte n automates, on dit que la chaîne est de longueur n. Une chaîne d'automates de longueur 3 est présentée sur la figure 2 avec la représentation classique. Pour simplifier, les étiquettes de transition ne sont pas représentées.

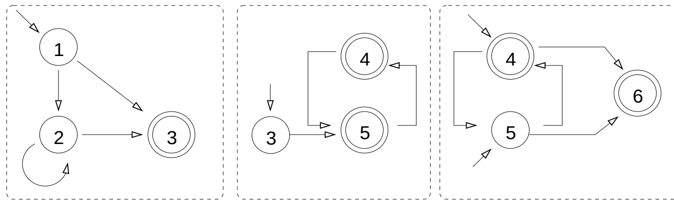

Fig. 2. Chaîne de trois automates

Définition 9 (Concaténation de chaîne) *Soit* $\mathcal{E}_A = (A^1, \ldots, A^n)$*, une chaine d'automates avec* $A^i = (Q^i, E^i, T^i, I^i, F^i)$*. La concaténation de la chaîne d'automates notée* $\oplus\mathcal{E}_A$ *est un automate* $A' = (Q', E', T', I', F')$ *défini comme suit :*

- $Q' = (Q^1 \cup \ldots \cup Q^n) \times \{\mathcal{W}^1, \ldots, \mathcal{W}^n\}$,
- $E' = (E^1 \cup \ldots \cup E^n) \cup \{\Upsilon_1, \ldots, \Upsilon_{n-1}\}$,
- $T' = \{((q, \mathcal{W}^i), l, (q', \mathcal{W}^i)) \mid (q, l, q') \in T^i\} \cup \{((q, \mathcal{W}^i), \Upsilon_i, (q, \mathcal{W}^{i+1})) \mid q \in F^i \wedge q \in I^{i+1}\}$,
- $I' = I^1 \times \{\mathcal{W}^1\}$,
- $F' = F^n \times \{\mathcal{W}^n\}$.

Puisque les automates d'une chaîne d'automates peuvent avoir des états en commun, le fait d'être dans un état q ne permet pas forcément de connaître la fenêtre temporelle courante. Aussi, nous qualifions ces états de *relatifs*. À l'inverse, nous appelons *absolus* les états $(q, \mathcal{W}^i)$ obtenus par concaténation d'une chaîne.

La concaténation consiste à transformer les états relatifs en états absolus et reconstruire les transitions. La concaténation de la chaîne d'automates de la figure 2 est représentée sur la figure 3. Par souci de lisibilité, les états $(q, \mathcal{W}^i)$ sont représentés (q, i).

Par abus de notation, on pourra écrire : $\oplus\mathcal{E}_A = A^1 \oplus \ldots \oplus A^n = (A^1 \oplus \ldots \oplus A^{n-1}) \oplus A^n$.

Nous montrons maintenant comment nous représentons un automate par une chaîne d'automates. Pour cela, il est tout d'abord nécessaire de revenir sur les trajectoires.

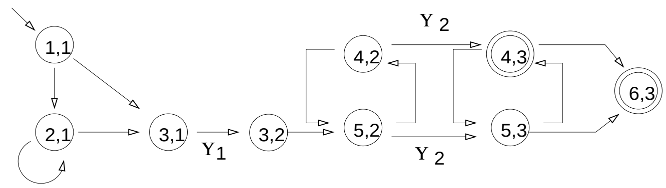

Fig. 3. Concaténation de la chaîne présentée sur la figure 2

Définition 10 (Abstraction de trajectoire) *Soit une trajectoire sur des états absolus* $traj' = ((q'_0, \ldots, q'_n), (l'_1, \ldots, l'_n))$*. Alors, l'abstraction* $traj$ *de la trajectoire* $traj'$ *est définie par* $traj = ((q_0, \ldots, q_n), (l_1, \ldots, l_n))$ *telle que :*
- $\forall i \in \{0, \ldots, n\},\ \exists k,\ q'_i = (q_i, \mathcal{W}^k)$,
- $\forall i \in \{1, \ldots, n\},\ (\exists k, l'_i = \Upsilon_k \Rightarrow l_i = \emptyset) \vee (\nexists k, l'_i = \Upsilon_k \Rightarrow l_i = l'_i)$.

Définition 11 (Abstraction d'automate) *Soit* $A' = (Q', E', T', I', F')$ *un automate sur des états absolus et* $A = (Q, E, T, I, F)$ *un automate sur des états relatifs. On dit que* A *est une* abstraction *de* A'*, noté* $A \simeq_{abs} A'$*, ssi :*
- *pour toute trajectoire* $traj$ *de* A*, il existe une trajectoire* $traj'$ *de* A' *et une trajectoire* $traj_2$ *de* A *telles que* $traj_2 = traj$ *et* $traj_2$ *est l'abstraction de* $traj'$*,*
- *pour toute trajectoire* $traj'$ *de* A'*, il existe une trajectoire* $traj$ *de* A *telle que* $traj$ *est l'abstraction de* $traj'$*.*

Définition 12 (Reconstruction) *Soit* $\mathcal{E}_A$ *une chaîne d'automates. Soit* $A = (Q, E, T, I, F)$*, un automate tel que* $\forall k,\ \Upsilon_k \notin E$*. On dit que* A *est une reconstruction de* $\mathcal{E}_A$ *(noté* $A \simeq_{rec} \mathcal{E}_A$*) ssi* A *est une abstraction de* $\oplus\mathcal{E}_A$*.*

Définition 13 (Découpage) *Soit* A *un automate et* $\mathcal{E}_A$ *une chaine d'automates. On dit que* $\mathcal{E}_A$ *est un découpage de* A *ssi* A *est une reconstruction de* $\mathcal{E}_A$*.*

L'automate présenté sur la figure 4 est l'abstraction de l'automate de la figure 3. On peut donc dire que la chaîne d'automates de la figure 2 est un découpage de l'automate de la figure 4.

Propriété 1 *Soit* A *un automate. Soit* $\mathcal{E}_A$ *un découpage de* A*. Soit* $A_2 = (Q_2, E_2, T_2, I_2, F_2)$*, un automate tel que* $\forall k, \Upsilon_k \notin E_2$*. Alors,* $A \otimes A_2 \simeq_{abs} (\oplus\mathcal{E}_A) \otimes A_2$*.*

4.3 Synchronisation de chaînes d'automates

Définition 14 (Automate prefix-closed) *Soit* $A = (Q, E, T, I, F)$*, un automate. On appelle* automate prefix-closed *de* A*, (noté* A^+*), l'automate* A *dont tous les états sont finaux (*$F^+ = Q$*).*

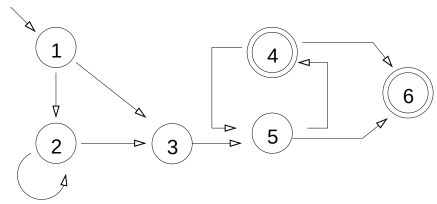

Fig. 4. Abstraction de l'automate Figure 3

Définition 15 (Automate suffix-closed) *Soit $A = (Q, E, T, I, F)$, un automate. On appelle* automate suffix-closed de A, *(noté A^-), l'automate A dont tous les états sont initiaux ($I^- = Q$).*

On note $A^{\#}$, l'automate prefix-closed, suffix-closed ($A^{\#} = A^{+^-} = A^{-^+}$).

Définition 16 (Synchronisation de chaîne avec un automate) *On appelle* synchronisation d'une chaîne $\mathcal{E}_A = (A^1, \ldots, A^n)$ avec un automate A_2 *la chaîne notée $\mathcal{E}_A \otimes A_2$ définie par :*
$\mathcal{E}_A \otimes A_2 = (A^1 \otimes A_2^+, A^2 \otimes A_2^{\#}, \ldots, A^{n-1} \otimes A_2^{\#}, A^n \otimes A_2^-)$.

La synchronisation d'une chaîne d'automates avec un automate A_2 consiste à synchroniser chacun des automates de la chaîne avec l'automate A_2. Il est nécessaire de synchroniser l'automate $A^k, k \neq n$ avec l'automate prefix-closed A_2^+ parce que l'état de l'automate A_2 au top Υ_k n'est pas forcément un état final de A_2. De même, il convient d'utiliser l'automate suffix-closed pour $k \neq 1$.

Propriété 2 *Soit $\mathcal{E}_A$ une chaîne d'automates et $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ tel que $\Upsilon \notin E_2$. Alors, $\oplus(\mathcal{E}_A \otimes A_2) = (\oplus\mathcal{E}_A) \otimes A_2$.*

Cette propriété peut être aisément montrée en vérifiant l'égalité de l'ensemble des trajectoires de $\oplus(\mathcal{E}_A \otimes A_2)$ et $(\oplus\mathcal{E}_A) \otimes A_2$.

Corollaire 1 *Soit A un automate. Soit $\mathcal{E}_A$ un découpage de A. Soit $A_2 = (Q_2, E_2, T_2, I_2, F_2)$, un automate tel que $\forall k, \Upsilon_k \notin E_2$. Alors, $\mathcal{E}_A \oplus A_2$ est un découpage de $A \otimes A_2$.*

4.4 Diagnostic

Soit OBS_n l'automate représentant les observations envoyées par le système et $\mathcal{E}_{OBS_n}$ un découpage de OBS_n.

Soit $MOD = (Q^{MOD}, E^{MOD}, T^{MOD}, I^{MOD}, F^{MOD})$ le modèle du système. Remarquons que MOD ne fournit aucune information sur l'état final du système.

On a donc la propriété suivante : $F^{MOD} = Q^{MOD}$. Ainsi, $MOD^{+} = MOD$ et $MOD^{\#} = MOD^{-}$.

On a alors :

$$\Delta_n \simeq_{rec} \mathcal{E}_{OBS_n} \otimes MOD \tag{2}$$

5 Diagnostic incrémental

On désire à présent calculer le diagnostic de manière incrémentale. Nous désirons donc calculer le diagnostic de la fenêtre $\mathcal{W}_{i+1}$ à partir du diagnostic de la fenêtre $\mathcal{W}_i$ et des informations sur la fenêtre $\mathcal{W}^{i+1}$.

Tout d'abord, nous justifions l'utilisation de chaînes de diagnostics à la place du diagnostic, puis nous présentons une solution issue des résultats de la section précédente. Après cela, nous donnons une amélioration du calcul.

5.1 Diagnostic sous forme de chaîne

Comme nous l'avons présenté au début de ce papier, nous définissons le diagnostic comme étant le calcul des trajectoires sur le modèle. Nous avons donc décidé de représenter le comportement du système par un automate. Cependant, une chaîne d'automates peut également représenter un ensemble de trajectoires.

Définition 17 (Concaténation de trajectoires) *Soit i trajectoires $traj^k = ((q_0^k, \ldots, q_{n(k)}^k), (l_1, \ldots, l_{n(k)}))$ telles que $\forall k \in \{1, \ldots, i-1\}$, $q_{n(k)}^k = q_0^{k+1}$. Alors, $traj$, la concaténation des i trajectoires $traj^k$, est définie par :*
$traj = ((q_0^1, \ldots, q_{n(0)}^1, \ldots, q_0^i, \ldots, q_{n(i)}^i), (l_1^1, \ldots, l_{n(1)}^1, \ldots, l_1^i, \ldots, l_{n(i)}^i))$.

Propriété 3 *Soit A un automate, et $\mathcal{E}_A = (A^1, \ldots, A^i)$ un découpage de A de longueur i.*

- *Soit $\forall k \in \{1, \ldots, i\}$, $traj^k = ((q_0^k, \ldots, q_{n(k)}^k), (l_1, \ldots, l_{n(k)}))$, i trajectoires sur les automates A^k. Alors, la concaténation des i trajectoires, si elle existe, est une trajectoire de A.*
- *Soit $traj$ une trajectoire de A. Alors, il existe i trajectoires $traj^k$ sur les automates A^k telles que la concaténation des i trajectoires est $traj$.*

Cette propriété est une extension logique des précédentes propriétés. On voit donc qu'une chaîne d'automates représente l'ensemble des trajectoires de l'automate calculé par reconstruction de cette chaîne.

5.2 Première méthode

Soit $\mathcal{E}_{\Delta i} = (\Delta^1, \ldots, \Delta^i)$, le diagnostic du système calculé incrémentalement pendant la fenêtre $\mathcal{W}_i$. Soit $\mathcal{E}_{\Delta i+1}$ le diagnostic du système pendant la fenêtre $\mathcal{W}_{i+1}$. Alors, on a :

$$\mathcal{E}_{\Delta i+1} = (\Delta^1, \ldots, \Delta^i, \Delta^{i+1}) \quad \text{avec } \Delta^{i+1} = OBS^{i+1} \otimes MOD^{-} \tag{3}$$

Ce résultat provient du fait que $MOD^{\#} = MOD^{-}$ (car tous les états sont finaux, voir la définition 5 du modèle). Si l'ensemble des états de MOD est grand, alors le nombre d'état initiaux de MOD^{-} est important et le calcul du diagnostic local à la fenêtre $\mathcal{W}^{i+1}$ est très coûteux. Il est donc nécessaire de réduire ce coût en limitant le nombre d'états initiaux possibles du système pour chaque fenêtre temporelle. C'est l'objectif de cette deuxième méthode appelée synchronisation incrémentale.

5.3 Synchronisation incrémentale

Définition 18 (Restriction) *Soit $A = (Q, E, T, I, F)$, un automate. On appelle* restriction *de l'automate A sur l'ensemble initial I', noté $A[I']$, l'automate $A' = (Q, E, T, I \cap I', F)$.*

Définition 19 (Synchronisation incrémentale) *On appelle* synchronisation incrémentale *de la chaîne d'automates $\mathcal{E}_A = (A^1, \dots, A^n)$ par l'automate A_2, la chaîne $\mathcal{E}_B$ notée $\mathcal{E}_A \odot A_2$ et définie ainsi : $\mathcal{E}_B = (A^1_B, \dots, A^n_B)$ avec $A^i_B = (Q^i_B, E^i_B, T^i_B, I^i_B, F^i_B)$ et :*
- $A^1_B = A^1 \otimes A_2^{+}$,
- $\forall i \in \{2, \dots, n-1\}$, $A^i_B = (A^i \otimes A_2^{\#})[F_B^{i-1}]$ *et*
- $A^n_B = (A^n \otimes A_2^{-})[F_B^{n-1}]$.

La synchronisation incrémentale consiste à reprendre les états finaux de la précédente fenêtre et de les utiliser comme états initiaux de la fenêtre courante.

Propriété 4 *Soit $\mathcal{E}_A$ une chaîne d'automates et $A_2 = (Q_2, E_2, T_2, I_2, F_2)$ tel que $\Upsilon \notin E_2$. Alors, $\oplus(\mathcal{E}_A \odot A_2) = \oplus(\mathcal{E}_A \otimes A_2)$.*

Notons $\Delta^i = (Q^i_\Delta, E^i_\Delta, T^i_\Delta, I^i_\Delta, F^i_\Delta)$. Soit $\mathcal{E}_{\Delta_i} = (\Delta^1, \dots, \Delta^i)$ le diagnostic du système pendant la fenêtre $\mathcal{W}_i$. Il est à présent possible de calculer le diagnostic incrémental de la manière suivante :

$$\mathcal{E}_{\Delta i+1} = (\Delta^1, \dots, \Delta^i, \Delta^{i+1}) \quad \text{avec } \Delta^{i+1} = (OBS^{i+1} \otimes MOD^{-})[F^i_\Delta] \tag{4}$$

On peut considérer que le nombre d'états finaux d'un diagnostic Δ^i est constant ou borné. Le calcul du diagnostic de la fenêtre $\mathcal{W}^{i+1}$ dépend alors de la taille de l'automate OBS^{i+1} (c'est-à-dire de la durée entre t_i et t_{i+1}). La complexité est donc suffisament faible pour pouvoir effectuer le calcul en-ligne.

6 Conclusion

Dans cet article, nous avons montré qu'il était possible de remplacer un automate par une chaîne d'automates. Nous avons défini les liens entre l'automate et la chaîne d'automates, et défini l'opération de synchronisation entre un automate et une chaîne d'automates.

Classiquement, le diagnostic est défini comme la synchronisation entre l'automate modélisant le système et l'automate des observations. Nous avons remplacé l'automate des observations par une chaîne d'automates, chacun des automates de la chaîne représentant les observations émises durant une fenêtre temporelle. Ceci nous a permis de calculer le diagnostic sous forme de chaîne de diagnostics locaux à une fenêtre temporelle. Nous avons montré que ce calcul était équivalent à un calcul de diagnostic global. De plus, nous avons montré qu'il était possible de calculer le diagnostic de manière incrémentale, c'est-à-dire en calculant les diagnostics locaux à chaque fenêtre temporelle successivement.

L'étape suivante de notre approche est le passage au diagnostic en-ligne. L'intérêt du diagnostic incrémental est qu'il sera possible de construire un diagnostic du système à un moment donné, puis d'utiliser le diagnostic de la dernière fenêtre temporelle pour continuer le diagnostic du système. Pour cela, il est nécessaire de formaliser la construction de la chaîne des observations dans le cadre du diagnostic en-ligne. Un problème est alors de savoir dans quelles conditions il est possible de découper en-ligne l'automate des observations en une chaine d'automates. La difficulté vient du fait qu'il faut vérifier que le découpage entamé sera correct alors que l'automate des observations n'est pas encore complet.

Références

1. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M. : Diagnosis of large active systems. Artificial Intelligence **110** (1999) 135–183
2. Cordier, M.O., Thiébaux, S. : Event-based diagnosis for evolutive systems. In : 5th International Workshop on Principles of Diagnosis (DX-94). (1994) 64–69
3. Barral, C., McIlraith, S., Son, T. : Formulating diagnostic problem solving using an action language with narratives and sensing. In : International Conference on Knowledge Representation and Reasoning (KR'2000). (2000) 311–322
4. Console, L., Picardi, C., Ribaudo, M. : Diagnosis and diagnosability analysis using PEPA. In : 14th European Conference on Artificial Intelligence (ECAI-00), Berlin, Allemagne (2000) 131–135
5. Cordier, M.O., Largouët, C. : Using model-checking techniques for diagnosing discrete-event systems. In : 12th International Workshop on Principles of Diagnosis (DX-2001). (2001) 39–46
6. Lamperti, G., Zanella, M. : Uncertain temporal observations in diagnosis. In : 14th European Conference on Artificial Intelligence (ECAI-00). (2000) 151–155
7. Lamperti, G., Zanella, M. : Diagnosis of Active Systems. Kluwer Academic Publishers (2003)
8. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D. : Failure diagnosis using discrete-event models. In : IEEE Transactions on Control Systems Technology (CST-96). (1996) 105–124
9. Rozé, L., Cordier, M.O. : Diagnosing discrete-event systems : extending the "diagnoser approach" to deal with telecommunication networks. Journal on Discrete-Event Dynamic Systems : Theory and Applications (JDEDS) (2002)