Motifs de surveillance pour le diagnostic de systèmes à événements discrets finis

Jéron Thierry, Marchand Hervé, Cordier Marie-Odile
IRISA, Campus universitaire de Beaulieu, Rennes
{Prénom.Nom}@irisa.fr

Résumé

Dans cet article, nous nous intéressons au diagnostic dans les systèmes de transition finis. Nous proposons un modèle de motifs de surveillance correspondant à des propriétés d'atteignabilité. Ceci permet de généraliser les propriétés à diagnostiquer tout en les découplant de la description du système. Nous en déduisons des techniques de vérification de diagnosticabilité et de construction de diagnostiqueur fondées sur des opérations standards sur les systèmes de transitions. Nous montrons que ces techniques sont suffisamment générales pour exprimer et résoudre de manière unifiée une classe importante de problèmes de diagnostic considérés dans la littérature comme le diagnostic de pannes permanentes, de pannes multiples, de séquences de pannes, et certains problèmes de diagnostic de pannes intermittentes.

Mots-Clé

Diagnosticabilité, motif de surveillance, systèmes de transition, atteignabilité.

Abstract

In this paper, we are interested in the diagnosis of finite transition systems. We propose a model of supervision patterns corresponding to reachability properties. This allows to generalize the properties to be diagnosed and to render them independent of the description of the system. We thus deduce techniques for the verification of the diagnosticability and the construction of a diagnoser based on standard operations on transition systems. We show that these techniques are general enough to express and solve in a unified way a broad class of diagnosis problems found in the literature, e.g. diagnosing permanent faults, multiple faults, fault sequences and some problems of intermittent faults.

Keywords

Diagnosticability, supervision pattern, transition systems, reachability.

1 Introduction

Depuis plusieurs années, de nombreuses recherches ont été menées autour du diagnostic, que ce soit pour des systèmes dynamiques, des réseaux de distribution ou des réseaux de télécommunications [11, 1, 9, 8, 3, 4].
L'une des approches les plus classiques pour le diagnostic de système à événements discrets traite du problème de la détection et de l'identification de motifs d'événements particuliers *a priori* inobservables dans les systèmes. Ces événements modélisent par exemple des défaillances, des fautes ou diverses anomalies dans le système. Ces motifs peuvent être simplement l'occurrence d'une panne permanente [10, 11], plusieurs occurrences d'une panne [7], la réparation du système après l'occurrence d'une panne [2], etc. Le but du diagnostic est alors, en se basant sur l'observation, de pouvoir déterminer de manière certaine, qu'un tel motif d'événements s'est produit, dans un délai borné après son occurrence. Suivant le motif à diagnostiquer, le comportement du système et le niveau d'observabilité du diagnostiqueur sur le système, le diagnostic n'est pas toujours possible mais la diagnosticabilité du système peut être vérifiée *a priori*.
Dans la littérature, les approches proposées manquent cependant de généricité. Une des raison est que les motifs d'événements à diagnostiquer sont rarement explicités de manière externe au système. Par conséquent on trouve une définition différente de la diagnosticabilité pour chaque motif. Les algorithmes de construction de diagnostiqueur, et de vérification de la diagnosticabilité qui en résultent [12, 6, 7, 11, 2] sont donc souvent ad hoc, difficilement vérifiables ou réutilisables pour d'autres problèmes.

Contribution et plan : La contribution de cet article vise à unifier ces problèmes de diagnostic et les algorithmes sous-jacents. On peut d'abord remarquer que la plupart des motifs de diagnostic traités dans la littérature, même si ils ne sont pas explicités, expriment des propriétés d'atteignabilité. En s'inspirant des techniques de model-checking, il est alors naturel d'exprimer ces motifs par un automate dans lequel l'ensemble d'états finals est stable (i.e. on ne peut en sortir), ce qui exprime le fait que tout prolonge-

ment d'une séquence acceptée est aussi acceptée, ou encore que le langage accepté est suffixe clos. Cet article propose une vue unifiée du diagnostic pour ce type de motifs de surveillance, se basant sur la donnée d'un automate décrivant les motifs à diagnostiquer de façon conjointe à la description du système. Les algorithmes de vérification de diagnosticabilité et de construction de diagnostiqueur qui en découlent sont alors fondés sur des opérations standards sur les automates (composition synchrone, ϵ-clôture, et déterminisation). Nous montrons que l'approche est alors suffisamment générale pour exprimer une classe importante de problèmes de diagnostic considérés dans la littérature, comme le diagnostic de pannes permanentes, de pannes multiples, de séquences de pannes, et certains problèmes de diagnostic de pannes intermittentes.
Après cette introduction, la section 2 présente les modèles de systèmes de transitions, les principales notations et opérations de base sur ces modèles et leurs propriétés. La section 3 introduit le problème de la diagnosticabilité de l'entrée dans un ensemble d'états stable, les algorithmes de vérification de la diagnosticabilité et de construction du diagnostiqueur correspondants. La section 4 traite du cas général du diagnostic pour des motifs de surveillance représentant des propriétés d'atteignabilité et montre comment ce problème se ramène au problème précédent. La fin de cette section illustre l'approche sur plusieurs types de motifs de surveillance traités dans la littérature, démontrant ainsi toute la généralité de notre approche. En conclusion nous donnons quelques perspectives d'extension de ce travail.

2 Préliminaires

2.1 Modèle des systèmes de transition

Le modèle que nous utilisons est celui des systèmes de transitions étiquetés, (en abrégé LTS pour *Labelled Transition System*).

Définition 1 (LTS) *Un LTS est un quadruplet* $G = (Q, \Sigma, \rightarrow, q_0)$ *où* Q *est un ensemble fini d'états,* $q_0 \in Q$ *est l'état initial,* Σ *est l'alphabet des actions (ou événements) et* $\rightarrow \subseteq Q \times \Sigma \times Q$ *est la relation de transition.*

Notations : Soit $G = (Q, \Sigma, \rightarrow, q_0)$ un LTS. On note $q \xrightarrow{\sigma} q' \triangleq (q, \sigma, q') \in \rightarrow$; cette notation est étendue à une séquence de la manière suivante : si $s = \sigma_1 \dots \sigma_n$, $q \xrightarrow{s} q' \triangleq \exists q_0, \dots, q_n : q = q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{a_n} q_n = q'$; On note $q \xrightarrow{s}$ pour $\exists q' : q \xrightarrow{s} q'$; l'ensemble des actions possibles en q est $\Gamma(q) \triangleq \{\sigma \in \Sigma \mid q \xrightarrow{\sigma}\}$; Un LTS G est dit *complet* si pour tout état q de G, $\Gamma(q) = \Sigma$. L'ensemble des états atteints à partir de q par la séquence s sera noté

$$\Delta_G(q, s) \triangleq \{q' \in Q \mid q \xrightarrow{s} q'\}$$

$\mathcal{L}(q) \triangleq \{s \in \Sigma^* \mid q \xrightarrow{s}\}$ est l'ensemble de séquences d'actions de G à partir de q ; le *langage généré* par le système G est $\mathcal{L}(G) \triangleq \mathcal{L}(q_0)$; étant donnée une séquence $s \in \mathcal{L}(G)$, l'ensemble des suffixes de s dans $\mathcal{L}(G)$ est noté

$$\mathcal{L}(G)/s \triangleq \{t \in \Sigma^* \mid s.t \in \mathcal{L}(G)\}$$

G peut jouer le rôle d'automate accepteur si on le munit d'un ensemble d'états finals Q_m ; le langage accepté de G dans Q_m est noté

$$\mathcal{L}(G, Q_m) = \{s \in \Sigma^* \mid \Delta_G(q_0, s) \subseteq Q_m\}.$$

G est *déterministe* si pour tout état $q \in Q$ et toute action $\sigma \in \Sigma$, $q \xrightarrow{\sigma} q' \wedge q \xrightarrow{\sigma} q'' \Rightarrow q' = q''$.
Un état q est *atteignable* s'il existe une séquence d'actions $s \in \Sigma^*$ telle que $q_0 \xrightarrow{s} q$. Le LTS G est *atteignable* si tous ses états sont atteignables. Il est *vivant* si $\forall q \in Q$, si q est atteignable, $\Gamma(q) \neq \emptyset$. Un ensemble d'états finals $Q_m \subseteq Q$ est dit *stable* pour $\rightarrow$ si $\rightarrow$ ne permet pas d'en sortir, i.e. toute prolongation dans G d'une séquence de G acceptée en Q_m est acceptée. Formellement

$$\forall q \in Q_m,\ \forall \sigma \in \Sigma, \Delta_G(q, \sigma) \in Q_m. \tag{1}$$

On suppose que l'ensemble des actions Σ est partitionné en deux sous-alphabets disjoints Σ_{uo} et Σ_o où Σ_o est l'ensemble des actions *observables* et Σ_{uo} l'ensemble des actions *inobservables*.
On note $P_{\Sigma_o} : \Sigma^* \rightarrow \Sigma_o^*$ la *projection* sur les événements observables : $P_{\Sigma_o}(\epsilon) = \epsilon$ et $\forall s \in \Sigma^*, P_{\Sigma_o}(\sigma.s) = \sigma.P_{\Sigma_o}(s)$ si $\sigma \in \Sigma_o$ et $P_{\Sigma_o}(\sigma.s) = P_{\Sigma_o}(s)$ si $\sigma \in \Sigma_{uo}$. On étend $P_{\Sigma_o}(\cdot)$ aux langages, $P_{\Sigma_o} : \mathcal{P}(\Sigma^*) \rightarrow \mathcal{P}(\Sigma_o^*)$ par $\forall L \subseteq \Sigma^*, P_{\Sigma_o}(L) = \{P_{\Sigma_o}(s) \mid s \in L\}$. Le *langage des traces* de G, défini par $Trace(G) = P_{\Sigma_o}(\mathcal{L}(G))$, représente donc le comportement observable du système G. On note aussi $Trace(G, Q_m) = P_{\Sigma_o}(\mathcal{L}(G, Q_m))$ l'ensemble des traces projetées de séquences acceptées.
Inversement, on définit la *projection inverse* $P_{\Sigma_o}^{-1} : \mathcal{P}(\Sigma_o^*) \rightarrow \mathcal{P}(\Sigma^*)$ tel que pour un langage $K \subseteq \Sigma_o^*$, $P_{\Sigma_o}^{-1}(K) = L \iff P_{\Sigma_o}(L) = K$. Intuitivement, $P_{\Sigma_o}^{-1}(K)$ est obtenu en insérant de toutes les manières possibles des séquences d'inobservables de $(\Sigma_{uo})^*$ dans les mots de K. Pour un langage $K \subseteq \Sigma_o^*$, on notera $P_G^{-1}(K) \triangleq P_{\Sigma_o}^{-1}(K) \cap \mathcal{L}(G)$ (i.e. pour une trace $\mu \in \Sigma_o^*$, $P_G^{-1}(\mu)$ est l'ensemble des séquences du système G équivalentes à s du point de vue de l'observation).

2.2 Opérations sur les systèmes de transitions

Produit synchrone : Le produit synchrone de deux LTS est une opération usuelle pour intersecter les langages générés et acceptés de ces deux LTS.

Définition 2 *Soient $G^i = (Q^i, q_0^i, \Sigma, \rightarrow_i)$, $i = 1, 2$, deux LTS. Leur produit synchrone $G^1 \times G^2$ est le LTS $G = (Q, q_0, \Sigma, \rightarrow)$ où $q_0 = (q_0^1, q_0^2)$, et où $Q \subseteq Q^1 \times Q^2$ et $\rightarrow \subseteq Q \times \Sigma \times Q$ sont obtenus par les règles suivantes :*

$$\cdot \quad q_0 = (q_0^1, q_0^2) \in Q$$

$$\cdot \quad \frac{(q^1, q^2) \in Q,\ q^1 \xrightarrow{\sigma}_1 q'^1,\ q^2 \xrightarrow{\sigma}_2 q'^2}{(q'^1, q'^2) \in Q,\ (q^1, q^2) \xrightarrow{\sigma} (q'^1, q'^2)}$$

Le langage du produit satisfait $\mathcal{L}(G) = \mathcal{L}(G^1) \cap \mathcal{L}(G^2)$. De plus, si pour $i = 1, 2$, G^i est muni de l'ensemble d'états finals Q_{m_i}, et G est muni de $Q_m = Q_{m_1} \times Q_{m_2}$, on a $\mathcal{L}(G, Q_m) = \mathcal{L}(G^1, Q_{m_1}) \cap \mathcal{L}(G^2, Q_{m_2})$. Notons aussi que si pour $i = 1, 2$, Q_{m_i} est stable pour $\rightarrow_i$, Q_m est stable pour $\rightarrow$.

Élimination des événements inobservables : L'ϵ-clôture d'un LTS consiste à éliminer ses actions inobservables tout en conservant les traces. Cette opération sera centrale dans la vérification de la diagnosticabilité.

Définition 3 (ϵ-clôture) *Étant donné un LTS $G = (Q, \Sigma, \rightarrow, q_0)$, avec $\Sigma = \Sigma_{uo} \cup \Sigma_o$, l'$\epsilon$-clôture de G, est le LTS $\epsilon(G) = (Q_\epsilon, \Sigma_o, \rightarrow_\epsilon, q_0)$, où $Q_\epsilon \subseteq Q$ et $\rightarrow_\epsilon \subseteq Q \times \Sigma_o \times Q$ sont définis par les règles suivantes :*

$$\cdot \quad q_0 \in Q_\epsilon$$

$$\cdot \quad \frac{q \in Q_\epsilon,\ q' \in Q,\ s \in \Sigma_{uo}^*,\ \sigma \in \Sigma_o, q \xrightarrow{s.\sigma} q'}{q' \in Q_\epsilon, q \xrightarrow{\sigma}_\epsilon q'}$$

On peut montrer que l'ϵ-clôture préserve les traces, i.e. $\mathcal{L}(\epsilon(G)) = Trace(\epsilon(G)) = Trace(G)$. Notons qu'on a choisi ici une définition de l'ϵ-clôture qui correspond à une clôture par actions inobservables suivies d'une action observable.

Déterminisation : L'opération de déterminisation consiste à calculer un LTS sans action inobservable et déterministe générant $Trace(G)$. Nous en donnons une définition directe depuis G, mais il peut se construire depuis $\epsilon(G)$ par la construction habituelle de sous-ensembles. La déterminisation sera utilisée dans la construction du diagnostiqueur.
Soit $x \subseteq Q$ un ensemble d'états, pour $\sigma \in \Sigma_o$, on note x after $\sigma \triangleq \{\Delta_G(q, \mu.\sigma) \mid q \in x, \mu \in \Sigma_{uo}^*\}$ l'ensemble des états atteignables depuis x par une séquence d'événements inobservables suivie de l'événement observable σ.

Définition 4 (Déterminisation) *Étant donné un LTS $G = (Q, \Sigma, \rightarrow, q_0)$, avec $\Sigma = \Sigma_{uo} \cup \Sigma_o$, le déterminisé de G est le LTS $Det(G) = (\mathcal{X}, \Sigma_o, \rightarrow_d, x_0)$, avec $x_0 = \{q_o\}$ et $\rightarrow_d$ et $\mathcal{X} \subseteq \mathcal{P}(Q)$ définis par les règles :*

$$\cdot \quad x_0 = \{q_o\} \in \mathcal{X}$$

$$\cdot \quad \frac{x \in \mathcal{X},\ \sigma \in \Sigma_o,\ x' = x \ \ after\ \sigma}{x' \in \mathcal{X},\ x \xrightarrow{\sigma}_d x'}$$

On a $Trace(G) = \mathcal{L}(Det(G))$. Notons que si $\mu \in Trace(G)$, et $\mu \neq \epsilon$, la relation suivante caractérise $\mathcal{X}$ en fonction de Q.

$$\Delta_{Det(G)}(x_0, \mu) = \{\Delta_G(q_0, s) \mid s \in P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o\} \quad (2)$$

Exemple 1 *Pour illustrer ces deux dernières opérations, considérons le LTS G donné en figure 1.*

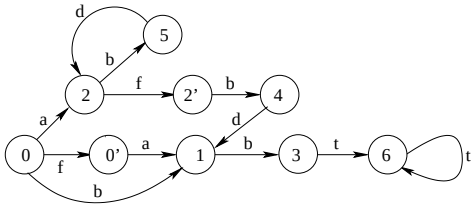


FIG. 1 – G

Supposons que seul l'événement f soit inobservable, $\epsilon(G)$ et $Det(G)$ sont alors donnés par les LTS de la Figure 2.

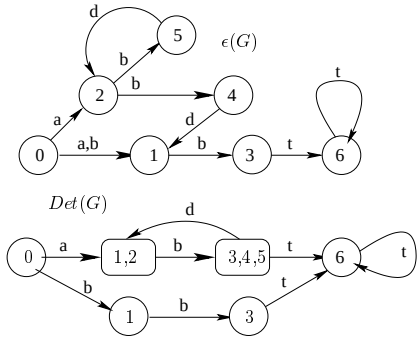


FIG. 2 – $\epsilon(G)$ et $Det(G)$

3 Diagnostic d'entrée dans un ensemble d'états stables

ous supposons donné un LTS muni d'états finals, dont les comportements représentent les comportements du système parmi lesquels ceux menant aux états finals sont les comportements à diagnostiquer. L'ensemble d'états finals est stable ce qui signifie que la sortie de l'ensemble d'états finals est impossible, en d'autres termes que la propriété à diagnostiquer est une propriété d'atteignabilité. Nous verrons par la suite comment obtenir ce LTS par produit synchrone entre un LTS du système et un automate décrivant les motifs à diagnostiquer.

3.1 Définition du modèle de pannes

Dans cet article, nous considérons un seul mode de panne, ce qui revient à se donner un seul ensemble d'états finals Q_P dont les éléments sont non distingués. La généralisation à n modes de pannes est facile.

Définition 5 *Un système à diagnostiquer est donné par le couple* (G, Q_P) *où*
- $G = (Q, \Sigma, \rightarrow, q_0)$, $\Sigma = \Sigma_{uo} \cup \Sigma_o$ *est un LTS déterministe, vivant et atteignable et sans composante fortement connexe terminale d'actions inobservables,*
- $Q_P \subseteq Q$ *est un ensemble d'états finals stable pour* $\rightarrow$ *avec* $q_0 \notin Q_P$.

L'hypothèse que G n'a pas de composante fortement connexe terminale d'actions inobservables permet d'éviter les cas où une action inobservable ferait transiter dans Q_P sans qu'aucune observation ne suive et ne permette de le diagnostiquer. Combiné avec la vivacité, ceci implique que $Det(G)$ est aussi vivant.

3.2 Diagnosticabilité

Le but du diagnostic est de pouvoir déterminer de manière certaine que le système est entré dans Q_P en un nombre borné d'observations après que ceci ne se soit produit. Nous donnons ici la définition formelle de diagnosticabilité adaptée de celle de [10, 11].

Définition 6 *Un système* G *est* Q_P*-diagnosticable si*

$$\exists n \in \mathbb{N},\ \forall s \in \mathcal{L}(G, Q_P), \forall t \in \mathcal{L}(G)/s,\\ \|P_{\Sigma_o}(t)\| \geq n \ \Rightarrow P_G^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(G, Q_P)$$

Intuitivement, un système G est Q_P-diagnosticable si et seulement si après une séquence s du système reconnue par Q_P, $(s \in \mathcal{L}(G, Q_P))$, si on observe suffisamment d'événements $(t \in \mathcal{L}(G)/s, \|P_{\Sigma_o}(t)\| \geq n)$, alors toutes les séquences compatibles avec l'observation $(P_G^{-1}(P_{\Sigma_o}(s.t)))$ sont aussi reconnues par Q_P (appartiennent à $\mathcal{L}(G, Q_P)$). La figure 3 explique cette définition (les traits pointillés correspondent à des comportements dans Q_P).

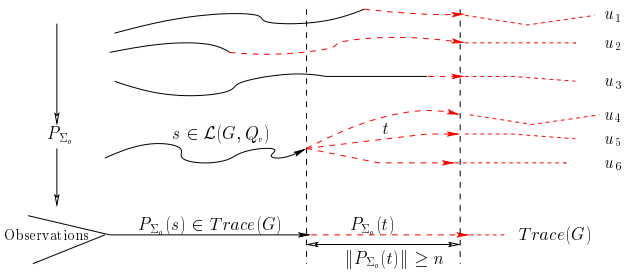


FIG. 3 – Idée intuitive de la diagnosticabilité

Remarque 1 *Si le système* G *ne possède aucun cycle d'actions inobservables, (comme supposé dans [10, 11]) il est équivalent de faire porter la borne* n *sur la longueur des séquences ou sur la longueur de l'observation. Notre définition est donc légèrement plus générale que celle de [10, 11].* ◇

3.3 Construction du diagnostiqueur.

Étant donné le caractère inobservable d'une partie des événements du système, le diagnostiqueur doit travailler sur une estimation de son état courant (caractérisée par un ensemble d'états possibles, dans lesquels le système peut avoir évolué après une trace donnée). Le diagnostiqueur est donc simplement un observateur externe au système qui, sur la base de ses observations, doit estimer si les séquences compatibles avec cette observation sont reconnues ou non par Q_P. Il est construit à partir de $Det(G)$ comme suit :

Définition 7 *Soit un système à diagnostiquer* (G, Q_P) *avec* $G = (Q, \Sigma, \rightarrow, q_0)$ *et* $Q_P \subseteq Q$.
Le diagnostiqueur de (G, Q_P) *est le LTS déterminisé* $G_d = Det(G) = (\mathcal{X}, \Sigma_o, x_o, \rightarrow_d)$, *muni de la fonction* DIAG : $\mathcal{X} \longrightarrow \{P, N, In\}$ *définie par*

$$\text{DIAG}(x) = \begin{cases} P & si \quad x \subseteq Q_P \\ N & si \quad x \cap Q_P = \emptyset \\ In & autrement \end{cases} \tag{3}$$

Un état $x \in \mathcal{X}$ est atteint par des observations $\mu \in \mathcal{L}(G_d, \{x\}) \subseteq \Sigma_o^*$. Pour toute observation $\mu \in \mathcal{L}(G_d, x) \setminus \{\epsilon\}$, notons $Comp(\mu) = P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o$ l'ensemble des séquences de G compatibles avec l'observation μ et se terminant par une observation. Intuitivement, le diagnostic P est donc émis quand toutes les séquences de $Comp(\mu)$ sont acceptées dans Q_P. Comme Q_P est stable, toute prolongation par des actions inobservables ne change pas l'acceptation en Q_P, donc ceci est équivalent au fait que toutes les séquences de $P_G^{-1}(\mu)$ sont reconnues par Q_P.
Le diagnostic N est émis quand aucune des séquences de $Comp(\mu)$ n'est reconnue par Q_P. Ici on ne peut pas s'abstraire de l'intersection avec $\Sigma^*.\Sigma_o$, car il est possible de transiter dans Q_P depuis $Q \setminus Q_P$ par des actions inobservables. Le diagnostic In est émis dans tous les autres cas, i.e. les cas où certaines séquences de $Comp(\mu)$ sont acceptées et d'autres non.
La proposition suivante formalise ces remarques :

Proposition 1 $\forall \mu \in \mathcal{L}(G_d)$,

$$\Delta_{G_d}(x_0, \mu) \subseteq Q_P \iff P_G^{-1}(\mu) \subseteq \mathcal{L}(G, Q_P) \tag{4}$$

$$\Delta_{G_d}(x_0, \mu) \cap Q_P = \emptyset \\ \iff P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o \cap \mathcal{L}(G, Q_P) = \emptyset \tag{5}$$

Preuve : On traite les cas $\mu = \epsilon$ et $\mu \neq \epsilon$ séparément. Si $\mu = \epsilon$, on a $\Delta_{G_d}(x_0, \epsilon) = x_0 \not\subseteq Q_P$ et $\epsilon \in P_G^{-1}(\epsilon)$ or $\epsilon \notin \mathcal{L}(G, Q_P)$ ce qui montre (4) pour $\mu = \epsilon$. par ailleurs, on a $\Delta_{G_d}(x_0, \epsilon) \cap Q_P = \emptyset$ et $P_G^{-1}(\epsilon) \cap \Sigma^*.\Sigma_o = \emptyset$ donc $P_G^{-1}(\epsilon) \cap \Sigma^*.\Sigma_o \cap \mathcal{L}(G, Q_P) = \emptyset$ ce qui montre (5) pour $\mu = \epsilon$.
Si $\mu \neq \epsilon$, μ se décompose en $\mu_1.\sigma, \mu_1 \in \Sigma_o^*, \sigma_o \in \Sigma_o$.

Montrons d'abord (4). Posons $x = \Delta_{G_d}(x_0, \mu)$ et supposons $x \subseteq Q_{\text{P}}$. D'après la construction de G_d (voir (2)), $x = \{\Delta_G(q_0, s) \mid s \in P_G^{-1}(\mu_1).\sigma_o\}$ donc $\forall s \in P_G^{-1}(\mu_1).\sigma_o, \Delta_G(q_0, s) \in Q_{\text{P}}$. Comme Q_{P} est stable, $\forall s' \in P_G^{-1}(\mu_1).\sigma_o.\Sigma_{uo}^* \cap \mathcal{L}(G)$ on a encore $\Delta_G(q_0, s') \in Q_{\text{P}}$. Or $P_G^{-1}(\mu) = P_G^{-1}(\mu_1).\sigma_o.\Sigma_{uo}^* \cap \mathcal{L}(G)$ donc $P_G^{-1}(\mu) \subseteq \mathcal{L}(G, Q_{\text{P}})$.
Inversement $P_G^{-1}(\mu) \subseteq \mathcal{L}(G, Q_{\text{P}})$, implique $P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o \subseteq \mathcal{L}(G, Q_{\text{P}})$ donc $\forall s \in P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o, \Delta_G(q_0, s) \subseteq Q_{\text{P}}$. Or d'après (2), $\Delta_{G_d}(x_0, \mu) = \{\Delta_G(q_0, s) \mid s \in P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o\}$ d'où $\Delta_{G_d}(x_0, \mu) \subseteq Q_{\text{P}}$.
Montrons maintenant (5). Posons $x = \Delta_{G_d}(x_0, \mu)$ et supposons $x \cap Q_{\text{P}} = \emptyset$. D'après la construction de G_d (voir (2)), $x = \{\Delta_G(q_0, s) \mid s \in P_G^{-1}(\mu_1).\sigma_o\}$ donc $\forall s \in P_G^{-1}(\mu_1).\sigma_o, \Delta_G(q_0, s) \notin Q_{\text{P}}$. Or $P_G^{-1}(\mu_1).\sigma_o = P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o$ donc $P_G^{-1}(\mu) \cap \Sigma^*\Sigma_o \cap \mathcal{L}(G, Q_{\text{P}}) = \emptyset$
Inversement si $P_G^{-1}(\mu) \cap \Sigma^*\Sigma_o \cap \mathcal{L}(G, Q_{\text{P}}) = \emptyset$, d'après (2), $\Delta_{G_d}(x_0, \mu) = \{\Delta_G(q_0, s) \mid s \in P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o\}$. Donc $\Delta_{G_d}(x_0, \mu) \cap Q_{\text{P}} = \emptyset$. ◇

Il reste à montrer que si le système est diagnosticable, le diagnostiqueur construit en Définition 7 est correct. Ceci est établi par la Proposition 2 qui dit que si une séquence du système pénètre Q_{P}, le diagnostiqueur construit nous l'indiquera en temps fini (i.e. après l'occurrence d'un nombre fini d'observations).

Proposition 2 *Soit $G = (Q, \Sigma, \rightarrow, q_0, Q_{\text{P}})$ un système à diagnostiquer et son diagnostiqueur $G_d = Det(G)$ muni de sa fonction* Diag. *Si G est Q_{P}-diagnosticable alors*

$$\exists n \in \mathbb{N}, \forall s \in \mathcal{L}(G, Q_P), \forall t \in \mathcal{L}(G)/s, \\ \|P_{\Sigma_o}(t)\| \geq n \Rightarrow \text{Diag}(\Delta_{G_d}(x_o, P_{\Sigma_o}(s.t)) = P \quad (6)$$

Preuve : Rappelons que la Q_{P}-diagnosticabilité est définie par,

$$\exists n \in \mathbb{N},\ \forall s \in \mathcal{L}(G, Q_{\text{P}}), \forall t \in \mathcal{L}(G)/s, \\ \|P_{\Sigma_o}(t)\| \geq n\ \Rightarrow P_G^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(G, Q_{\text{P}}).$$

En appliquant la proposition 1 à $P_{\Sigma_o}(s.t)$ on a $P_G^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(G, Q_{\text{P}})$ si et seulement si $\Delta_{G_d}(x_0, P_{\Sigma_o}(s.t)) \subseteq Q_{\text{P}}$ ce qui équivaut à $Diag(\Delta_{G_d}(x_0, P_{\Sigma_o}(s.t))) = P$, ce qui termine la preuve. ◇

Exemple 2 *Pour illustrer la définition 7, considérons le LTS G (adapté de [10]) représenté en Figure 4. Supposons que dans G, $Q_{\text{P}} = \{0', 1, 2', 3, 4, 6\}$ ce qui correspond aux états atteignables après l'occurrence de la panne modélisée par l'événement f. Suivant la définition 7, les états $\{1, 2\}$ et $\{3, 4, 5\}$ de $\mathcal{X}$ sont indéterminés (i.e. étiquetées In) pour le diagnostic, alors que $\{6\}$ indique que le système est sûrement en panne.*

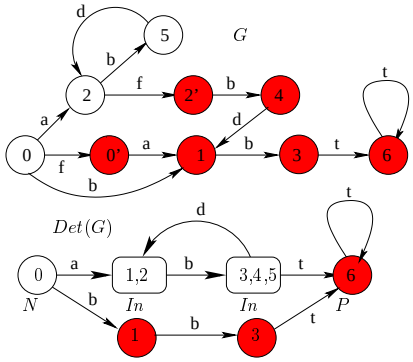


Fig. 4 – G et son diagnostiqueur associé

On peut toutefois constater que le diagnostiqueur seul ne permet pas de conclure que le système est diagnosticable ou non. Il existe un cycle d'états indéterminés entre $\{1, 2\}$ et $\{3, 4, 5\}$[1]*, qui laisse à penser que si l'on observe une séquence arbitrairement longue $a(bd)^n, n \in \mathbb{N}$, alors on ne saura jamais pas si le système est en panne ou non. Ce système est pourtant diagnosticable. En effet dès que le système évolue dans un état panne ($0'$ ou $2'$), alors l'événement t sera observé après un nombre d'événements borné par 4. L'observation de t indique donc de manière certaine que le système est passé par f.* ◇

Exemple 3 *A contrario, considérons l'exemple suivant donné en Figure 5. Ce système n'est pas diagnosticable car le cycle indéterminé dans le diagnostiqueur correspond cette fois-ci à de vrais cycles indéterminés dans le système, certains où f se produit, d'autres où il ne se produit pas. L'observation arbitrairement longue $a.(b.d)^n$ ne permet donc jamais de conclure, même si f s'est produit. On peut donc affirmer que l'information du diagnostiqueur est insuffisante pour savoir si le système est diagnosticable. Ceci a conduit à proposer des algorithmes de vérification de diagnosticabilité qui se basent, soit sur le diagnostiqueur et le système [11] soit sur $\epsilon(G)$ [6].* ◇

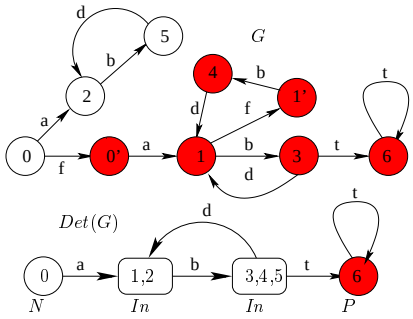


Fig. 5 – G et son diagnostiqueur associé

[1] i.e. $\text{Diag}(\{1, 2\}) = \text{Diag}(\{3, 4, 5\}) = In$

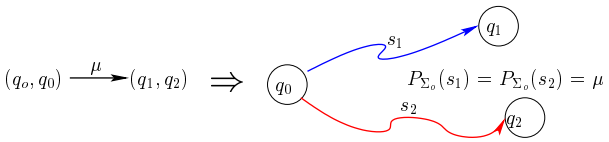
3.4 Vérification de la diagnosticabilité.

Un des aspects importants de la diagnosticabilité consiste à savoir *a priori* vérifier qu'un système est diagnosticable ou non. Les deux exemples précédents nous ont montré que le diagnostiqueur seul ne permet pas cette vérification (la présence d'un cycle indéterminé (i.e. étiqueté In) dans celui-ci n'implique pas forcément que le système soit non-diagnosticable). Rappelons que les hypothèses faites sur le système impliquent que $Det(G)$ est vivant. Cette hypothèse nous assure que quelque soit le comportement du système, il y aura toujours une observation possible. La méthode de vérification de diagnosticabilité de [12, 6] peut être adaptée à la diagnosticabilité de l'entrée dans un ensemble d'états stable. La preuve, adaptée de [6], est toutefois plus directe. L'idée intuitive de la vérification est que le système n'est pas diagnosticable lorsqu'il existe deux séquences de longueurs arbitrairement longues, équivalentes du point de vue de l'observation, l'une traversant des états de Q_{P}, l'autre non. Afin de pouvoir identifier de tels couples de séquences, nous introduisons le LTS suivant.
Soit (G, Q_P) un système à diagnostiquer et $\epsilon(G) = (Q_\epsilon, \Sigma_o, q_0, \rightarrow_\epsilon)$ l'ϵ-clôture de G. On note G_o le LTS défini par

$$\begin{array}{rcl} G_o & = & \epsilon(G) \times \epsilon(G) \\ & = & (\mathcal{X}_o, \Sigma_o, (q_o, q_o), \rightarrow_o), \text{ avec } \mathcal{X}_o = Q_\epsilon \times Q_\epsilon \end{array} \quad (7)$$

G_o va permettre d'identifier des couples de séquences équivalentes menant dans des états différents du système. En effet,

Lemme 1 *Soit G_o tel que défini en 7. Soit $\mu \in \mathcal{L}(G_o)$ tel que $(q_o, q_o) \xrightarrow{\mu}_o (q_1, q_2)$ et $q_1 \neq q_2$. Alors il existe deux séquences distinctes[2] $s_1, s_2 \in \mathcal{L}(G)$ tels que $q_o \xrightarrow{s_1} q_1$, $q_o \xrightarrow{s_2} q_2$ et $P_{\Sigma_o}(s_1) = P_{\Sigma_o}(s_2) = \mu$.*



Si de plus, q_1 est dans Q_{P} alors que q_2 n'est pas dans Q_{P} et que (q_1, q_2) est dans un cycle étiqueté par μ', alors pour tout l, la séquence $\mu.\mu'^l$ ne permet pas de diagnostiquer l'entrée dans Q_{P}. Nous allons montrer que l'absence de tels cycles est une condition nécessaire et suffisante à la diagnosticabilité.

Définition 8 *Soit $x = (q_1, q_2)$ un état de G_o. On dira que x est* indéterminé *si $q_1 \notin Q_{\text{P}}$ et $q_2 \in Q_P$ ou si $q_1 \in Q_P$ et $q_2 \notin Q_{\text{P}}$. Un cycle d'états Q_{P}-indéterminés dans G_o est une suite $(x_k, ..., x_n)$ t.q.*

$$x_k \xrightarrow{\sigma_k}_o x_{k+1} \cdots x_{n-1} \xrightarrow{\sigma_{n-1}}_o x_n \xrightarrow{\sigma_n}_o x_k$$

avec $\forall k \leq i \leq n, \sigma_i \in \Sigma_o$ et x_i est indéterminé.

La proposition suivante établit une condition nécessaire et suffisante de diagnosticabilité et donne un moyen algorithmique de la vérifier.

Proposition 3 *G est Q_{P}-diagnosticable ssi il n'existe pas de cycle d'états Q_P-indéterminés dans G_o.*

Preuve : Supposons qu'il existe un cycle indéterminé $(x_k, ..., x_n)$ dans G_o t.q. $x_k \xrightarrow{\sigma_k}_o x_{k+1} \cdots x_{n-1} \xrightarrow{\sigma_{n-1}}_o x_n \xrightarrow{\sigma_n}_o x_k$ avec $\sigma_i \in \Sigma_o$ et $\forall k \leq i \leq n, x_i = (q_i, q'_i)$ avec $q_i \notin Q_P$, $q'_i \in Q_P$. Notons $\rho = \sigma_k \cdots \sigma_n$. Comme G_o est atteignable, il existe $\mu \in \mathcal{L}(G_o)$ t.q. $\delta_o(x_o, \mu) = x_k$. De plus, comme $x_k = (q_k, q'_k)$ avec $q_k \neq q'_k$, d'après le lemme 1, $\exists s, s' \in \mathcal{L}(G)$, t.q. $q_o \xrightarrow{s} q_k$ et $q_o \xrightarrow{s'} q'_k$ avec $P_{\Sigma_o}(s) = P_{\Sigma_o}(s') = \mu$. Par ailleurs $s \notin \mathcal{L}(G, Q_{\text{P}})$ alors que $s' \in \mathcal{L}(G, Q_{\text{P}})$ Considérons maintenant la trace arbitrairement longue $\mu.\rho^l$. Par le lemme 1, il existe deux séquences distinctes $s.t$ et $s'.t'$ avec $P_{\Sigma_o}(t) = P_{\Sigma_o}(t') = \rho^l$, t.q. $q_k \xrightarrow{t} q_k$ et $q'_k \xrightarrow{t'} q'_k$. On a donc $s.t \notin \mathcal{L}(G, Q_{\text{P}})$, alors que $s'.t' \in \mathcal{L}(G, Q_{\text{P}})$. Donc G n'est pas Q_{P} diagnosticable.
Réciproquement, supposons qu'il n'existe aucun cycle indéterminé dans G_o et soit N la longueur de la plus grande séquence d'état indéterminés dans G_o. Supposons que G ne soit pas diagnosticable. Par définition, $\forall n$, il existe une séquence $s \in \mathcal{L}(G, Q_{\text{P}})$, un prolongement t suffisamment long $\|P_{\Sigma_o}(t)\| \geq n$, et une séquence u t.q. $P_{\Sigma_o}(u) = P_{\Sigma_o}(s.t)$ mais $u \notin \mathcal{L}(G, Q_{\text{P}})$. Posons $n = N + 2$.
Soit q le premier état de Q_{P} traversé par s dans G. Si q est atteint par un observable posons $q_1 = q$, sinon prenons pour q_1 le premier successeur de q atteint par un observable le long de $s.t$.
Soient $s_1, s_2 \in \Sigma^*$ t.q. $s.t = s_1.s_2$ et $q_o \xrightarrow{s_1} q_1$. Soit $u = u_1.\sigma_o.u_2$ avec $\sigma_o \in \Sigma_o$ t.q. $P_{\Sigma_o}(u_1.\sigma_o) = P_{\Sigma_o}(s_1)$,$P_{\Sigma_o}(u) = P_{\Sigma_o}(s.t)$ et q'_1 t.q. $q_o \xrightarrow{u_1.\sigma_o} q'_1$. Soit $x = (q_1, q'_1)$. On a $x_o \xrightarrow{P_{\Sigma_o}(u_1).\sigma_o}_o (q_1, q'_1)$ et (q_1, q'_1) est indéterminé.
Maintenant comme $s.t \in \mathcal{L}(G, Q_{\text{P}})$ et $u \notin \mathcal{L}(G, Q_{\text{P}})$, il existe une séquence d'états indéterminés initialisée en (q_1, q'_1) qui est traversée en tirant la trace $P_{\Sigma_o}(u_2) = P_{\Sigma_o}(s_2)$ et cette séquence d'états est de longueur supérieure à $N + 1$, ce qui contredit l'hypothèse de départ. Donc G est diagnosticable. ◇

Exemple 4 *Pour illustrer ce dernier point, considérons l'exemple 2. On suppose que $Q_{\text{P}} = \{0', 1, 2', 3, 4, 6\}$.*
Les tuples $\{(1,2), (2,1), (3,5), (5,3), (4,5), (5,4)\}$, dans G_o, sont indéterminés. Il est facile de voir qu'il n'y a aucun cycle d'états indéterminés. Par conséquent G est Q_{P}-diagnosticable. ◇

Remarque 2 *On peut noter que G_d et G_o sont indépendants de l'ensemble des états modélisant les*

[2] Si G était indéterministe, ce résultat ne serait plus vrai.

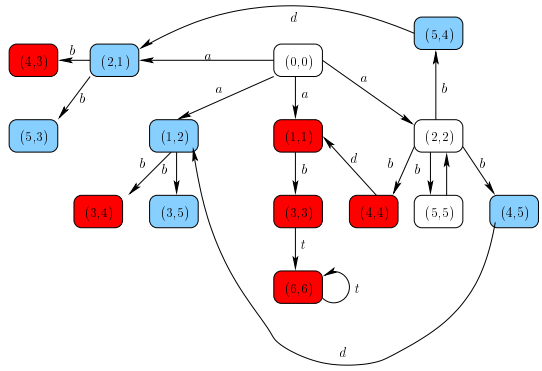


FIG. 6 – G_o pour le LTS de l'exemple 2 de la Figure 4

états de pannes. Changer cet ensemble ne nécessite pas de recalculer ces deux LTS. ◇

4 Diagnostic à base de motifs de surveillance

4.1 Cas général

Dans le diagnostic ou la surveillance, on s'intéresse à l'occurrence d'un événement ou à l'enchaînement de plusieurs événements particuliers, par exemple des pannes. Plus précisément, on s'intéresse à diagnostiquer, par l'observation du comportement externe observable du système, que le comportement interne du système satisfait une certaine propriété. Beaucoup de propriétés d'intérêt sont des propriétés d'atteignabilité, par exemple l'occurrence d'une panne, de la répétition d'une panne, l'occurrence de plusieurs pannes. Dans le cadre de la vérification de propriété sur les systèmes finis, une manière naturelle de décrire ces propriétés consiste à fournir un automate reconnaissant un langage sur l'alphabet des actions du système. Quand il s'agit de propriétés d'atteignabilité, l'ensemble des états finals de leurs automates est stable.
Nous introduisons maintenant la notion de motif de surveillance qui consiste en un tel automate :

Définition 9 *Un* motif de surveillance *est un LTS* $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega})$ *déterministe et complet, muni d'un ensemble d'états finals* $Q_F \subseteq Q_\Omega$ *stable et tel que* $q_{0_\Omega} \notin Q_F$.

Le problème de diagnostic auquel on s'intéresse est de savoir, sur la base d'observations d'un système G modélisé par un système de transitions $G = (Q, \Sigma, \rightarrow, q_0)$, si le système a effectué une séquence d'actions s acceptée par le motif de surveillance i.e. $s \in \mathcal{L}(\Omega, Q_F)$. Cela nous amène à la définition suivante de la diagnosticabilité.

Définition 10 *Un LTS* G *est* Ω*-diagnosticable si et seulement si*

$$\exists n \in \mathbb{N}, \forall s \in \mathcal{L}(\Omega, Q_F) \cap \mathcal{L}(G), \forall t \in \mathcal{L}(G)/s \\ \|P_{\Sigma_o}(t)\| \geq n \Rightarrow P_G^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(\Omega, Q_F)$$

Le résultat suivant montre qu'on peut ramener le problème de la Ω-diagnosticabilité à celui de la Q_P-diagnosticabilité :

Proposition 4 *Soit un LTS* $G = (Q, \Sigma, q_0, \rightarrow)$ *et un motif de surveillance* $\Omega = (Q_\Omega, \Sigma, q_{0_\Omega}, \rightarrow_\Omega)$ *muni de son ensemble d'états finals stable* $Q_F \subseteq Q_\Omega$.
G est Ω-diagnosticable si et seulement si $G \times \Omega$ est Q_P-diagnosticable pour $Q_P = Q \times Q_F$.

Preuve : Le résultat découle immédiatement des propriétés de $G_\Omega = G \times \Omega$. Rappelons que G_Ω est Q_P-diagnosticable si

$$\exists n \in \mathbb{N}, \ \forall s \in \mathcal{L}(G_\Omega, Q_P), \forall t \in \mathcal{L}(G_\Omega)/s, \\ \|P_{\Sigma_o}(t)\| \geq n \Rightarrow P_{G_\Omega}^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(G_\Omega, Q_P).$$

Le motif de surveillance étant un LTS Ω complet (donc $\mathcal{L}(\Omega) = \Sigma^*$), le langage généré du produit est $\mathcal{L}(G_\Omega) = \mathcal{L}(G) \cap \mathcal{L}(\Omega) = \mathcal{L}(G)$ et $Trace(G_\Omega) = Trace(G)$.
Par conséquent $\mathcal{L}(G_\Omega)/s = \mathcal{L}(G)/s$ et les fonction $P_{G_\Omega}^{-1}$ et P_G^{-1} sont identiques. Par ailleurs, pour l'ensemble d'états finals $Q_P = Q \times Q_F$ le langage accepté de G_Ω dans Q_P est $\mathcal{L}(G_\Omega, Q_P) = \mathcal{L}(G, Q) \cap \mathcal{L}(\Omega, Q_F) = \mathcal{L}(G) \cap \mathcal{L}(\Omega, Q_F)$. ◇

La Proposition 4 fournit une technique générale pour le diagnostic de propriétés d'atteignabilité décrit par un motif de surveillance Ω. En effet vérifier la Ω-diagnosticabilité pour l'automate et construire le diagnostiqueur se ramène à vérifier la Q_P-diagnosticabilité sur $G \times \Omega$ avec $Q_P = Q \times Q_F$ en utilisant la Proposition 3, puis construire le diagnostiqueur défini par la Définition 7. Les avantages de cette approche sont d'avoir une technique générale pour une classe importante de propriétés et, du point de vue méthodologique de dissocier la description du système de la description des propriétés à diagnostiquer.

4.2 Applications

Nous illustrons ici, pour des problèmes de diagnostic traités dans la littérature, comment décrire les motifs de surveillance permettant d'utiliser notre approche.

Occurrence d'une panne. Dans le cas du diagnostic d'une seule panne f, le motif de surveillance Ω est simplement l'automate O_f de la figure 7 avec $Q_F = \{F\}$. L'automate est déterministe, complet et Q_F est stable.

Exemple 5 *Pour le système G décrit par la figure 1, le produit $G_\Omega = G \times \Omega$ est le LTS représenté en haut de la Figure 4 où $Q_P = Q \times Q_F$. Le diagnostiqueur est représenté en bas de la Figure 4. On a vu que G_Ω était Q_P-diagnosticable, G est donc Ω-diagnosticable.* ◇

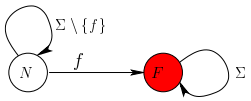


FIG. 7 – O_f : motif caractérisant la panne f

La f-diagnosticabilité est définie par [10, 11] pour un système G sans boucles d'actions inobservables par

$$\exists n \in \mathbb{N}, \forall s \in \Psi(f), \forall t \in \mathcal{L}(G)/_s,$$
$$\|t\| \geq n \Rightarrow \forall u \in \mathcal{L}(G), (P_{\Sigma_o}(u) = P_{\Sigma_o}(s.t)) \Rightarrow f \in u$$
$$\text{avec } \Psi(f) = \{s \in \mathcal{L}(G) \mid \exists s' \in \Sigma^*,\ s = s'f\}$$

On peut montrer la proposition suivant :

Proposition 5 *Pour un système G sans boucle d'actions inobservables et pour l'automate Ω décrit plus haut, la f-diagnosticabilité est équivalente à la Ω-diagnosticabilité*

Preuve : On rappelle la définition de Ω-diagnosticabilité :

$$\exists n \in \mathbb{N}, \forall s \in \mathcal{L}(\Omega, Q_F) \cap \mathcal{L}(G), \forall t \in \mathcal{L}(G)/s$$
$$\|P_{\Sigma_o}(t)\| \geq n \ \Rightarrow P_G^{-1}(P_{\Sigma_o}(s.t) \subseteq \mathcal{L}(\Omega, Q_F)$$

Dans le cas où G n'a pas de boucle d'actions internes, on peut remplacer $\|P_{\Sigma_o}(t)\| \geq n$ par $\|t\| \geq N$ avec $N = n * n_i$, où n_i est la plus grande séquence d'actions internes.
On a $s \in \Psi(f) \Rightarrow s \in \mathcal{L}(G, Q_F)$ et puisque Q_F est stable, $s \in \Psi(f)$ équivalent à $\forall s'' \in \mathcal{L}(G), (s'' = s.s' \Rightarrow s'' \in \mathcal{L}(G, Q_F))$. On peut donc remplacer $s \in \Psi(f)$ par $s \in \mathcal{L}(G, Q_F)$.
On a $f \in u$ équivalent à $u \in \mathcal{L}(\Omega, Q_F)$ donc $\forall u \in \mathcal{L}(G), (P_{\Sigma_o}(u) = P_{\Sigma_o}(s.t)) \Rightarrow f \in u$ équivaut à $\forall u \in \mathcal{L}(G), (P_{\Sigma_o}(u) = P_{\Sigma_o}(s.t)) \Rightarrow u \in \mathcal{L}(\Omega, Q_F)$ ou encore $P_G^{-1}(P_{\Sigma_o}(s.t)) \subseteq \mathcal{L}(\Omega, Q_F)$. Les deux définitions sont donc équivalentes dans le cas où G n'a pas de boucle d'actions internes. ◇

Occurrences de plusieurs types de pannes. Soient f_1 et f_2, deux pannes présentes dans le système. Et soient $\mathcal{O}_{f_1}$ et $\mathcal{O}_{f_2}$ les motifs de surveillance de f_1 et f_2. Pour considérer le problème du diagnostic de l'occurrence des pannes f_1 et f_2, il suffit de considérer le motif de surveillance de la Figure 8 obtenu pas produit synchrone entre $\mathcal{O}_{f_1}$ et $\mathcal{O}_{f_2}$ muni de $Q_F = (F_1, F_2)$ puisque celui-ci reconnaît le langage $\mathcal{L}(\mathcal{O}_{f_1}, F_1) \cap \mathcal{L}(\mathcal{O}_{f_2}, F_2)$. En présence de plusieurs pannes $\{f_1, \cdots, f_l\}$ à diagnostiquer, il est nécessaire de réaliser le produit des motifs de surveillance $\mathcal{O}_{f_i}$. La taille du motif de surveillance global est donc en $\mathcal{O}(2^l)$.
Si on s'intéresse à l'occurrence d'une panne parmi l types de pannes, le motif de surveillance sera l'automate qui reconnaît l'union des langages. Cet automate est de taille $\mathcal{O}(l)$.
Si on s'intéresse au diagnostic de *pannes en cascade*, avec un ordre sur l'occurrences des pannes,

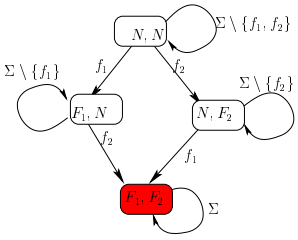


FIG. 8 – Motif de surveillance pour plusieurs types de pannes

par exemple f_2 après f_1, le motif de surveillance doit reconnaître la concaténation des langages $\mathcal{L}(\mathcal{O}_{f_1}, F_1).\mathcal{L}(\mathcal{O}_{f_2}, F_2)$, ce qui correspond au motif de surveillance représenté en Figure 9.

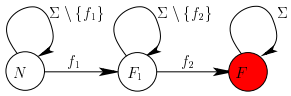


FIG. 9 – Pannes en cascade

Occurrences multiples de la même panne. Un autre problème de diagnostic est celui où on cherche à diagnostiquer la répétition d'un événement de panne f. Le motif de surveillance de la figure 10, qui reconnaît le langage $\mathcal{L}(\mathcal{O}_f, F)^k$ permet de diagnostiquer que la panne f s'est produite au moins k fois dans le système. Cela correspond à la k-diagnosticabilité de [7].

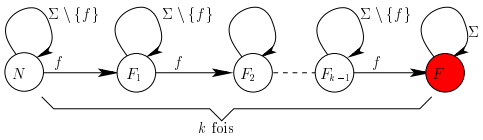


FIG. 10 – k occurrences de la panne f

On peut également considérer comme ensemble d'états pannes les ensembles $\{F_i, \cdots, F_{k-1}, F\}$, $i \leq k$. On obtient alors k motifs de surveillance imbriqués. On peut donc diagnostiquer par le même diagnostiqueur le fait que la panne s'est produite au moins i fois, pour $i \leq k$. Maintenant si l'on considère le diagnostiqueur qui rend vrai dès que l'un des motifs de surveillance précédents est diagnostiqué, alors cela correspond à la $[1-k]$-diagnosticabilité de [7].

Pannes intermittentes. Jusqu'à présent, nous avons considéré le cas de pannes permanentes. Or il existe de nombreux systèmes pour lesquels les pannes sont intermittentes (tout comme la panne peut être codée par un événement particulier, il est possible de faire de même pour les réparations de ces pannes). Nous donnons ici quelques exemples simples de problèmes de diagnostic de pannes avec réparations pour lesquels notre méthodologie peut être utilisée.

Le motif de surveillance de la Figure 11 décrit la propriété que la panne f s'est produite et qu'elle à été réparée au moins une fois (par contre au moment de ce diagnostic, rien ne dit que le système n'est pas de nouveau en panne). Ce type de diagnostic est appelé *diagnosticabilité de type I* dans [2].

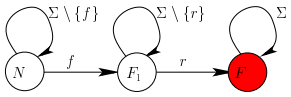


FIG. 11 – Panne intermittente avec réparation

Remarque 3 *Si l'on prend comme états finals de cet automate les états F_1 et F, on se contente de diagnostiquer le fait que la panne s'est produite dans le passé. On se retrouve alors dans cas de l'occurrence d'une panne. Ce type de diagnostic avec pannes intermittentes est appelé diagnosticabilité de type O dans [2].*

Le motif de surveillance de la Figure 12 permet de diagnostiquer le fait que la panne f s'est produite 2 fois sans avoir été réparée.

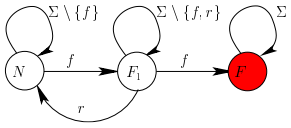


FIG. 12 – Panne intermittente avec réparation

Exemple de surveillance. L'exemple de la Figure 13 décrit un système avec boucles d'actions internes. Dans cet exemple, on modélise de manière simplifiée les déplacements d'une personne dans un bâtiment qui se composent d'un *Accueil* (A), d'un *Institut* (I), d'une *Bibliothèque* (B) et d'une *Cafétéria* (C). Les porte-couloirs allant d'un bâtiment à un autre ne peuvent s'emprunter que dans une seule direction et certains sont sécurisés par des cartes permettant l'accès mais aussi l'observation.

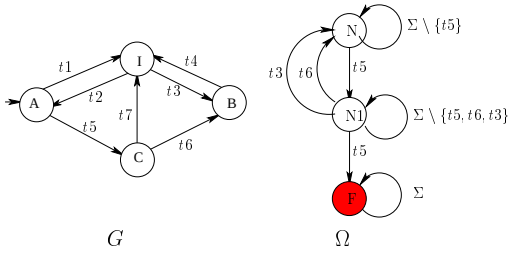


FIG. 13 – G et son motif de surveillance associé Ω

On suppose qu'il existe des capteurs sur les porte-couloirs t_1, t_2 et t_4 mais ces capteurs ne sont pas obligatoirement tous en service à un moment donné. On considère le modèle de surveillance Ω (Figure 13) qui exprime le fait qu'aller deux fois à la Cafétéria sans passer entre-temps à la Bibliothèque est un comportement à surveiller.
En suivant les étapes décrites dans la section précédente, le produit $G \times \Omega$ étiquette les états du système G en fonction du modèle de surveillance. Le LTS résultant est donné en Figure 14.

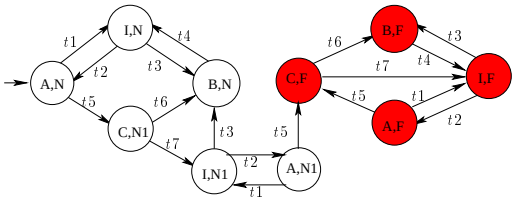


FIG. 14 – Système $G \times \Omega$ étiqueté par l'occurrence des pannes

On suppose dans un premier temps que seuls les événements t_1, t_2 sont observables, $\Sigma_o = \{t_1, t_2\}$. Le système observé (i.e. $\epsilon(G \times \Omega)$) est donné en Figure 15. Le LTS $G_o = \epsilon(G \times \Omega) \times \epsilon(G \times \Omega)$, (non construit ici)

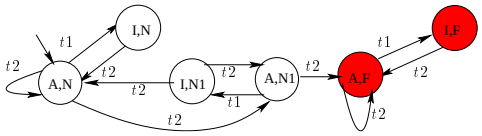


FIG. 15 – $\epsilon(G \times \Omega)$ pour $\Sigma_o = \{t_1, t_2\}$

présente un cycle indéterminé atteignable :
$((A,N),(A,N)) \xrightarrow{t_2}_o ((A,N),(A,N1)) \xrightarrow{t_2}_o$
$\cdots ((A,N),(A,F)) \xrightarrow{t_2} ((A,N),(A,F))$
donc G n'est pas diagnosticable pour $\Sigma_o = \{t_1, t_2\}$.
Par contre si l'on rajoute le capteur t_4 aux événements observables, $\Sigma_o = \{t_1, t_2, t_4\}$, le système observé $\epsilon(G \times \Omega)$ est alors donné par la Figure 16.

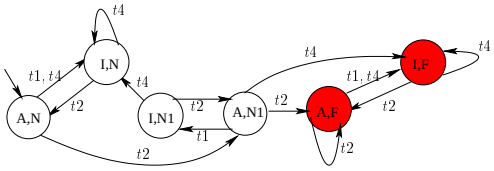


FIG. 16 – $\epsilon(G \times \Omega)$ pour $\Sigma_o = \{t_1, t_2, t_4\}$

Ce système est déterministe. Donc $G_o = \epsilon(G \times \Omega) \| \epsilon(G \times \Omega) = \epsilon(G \times \Omega)$ et G_o ne contient pas de cycle d'états indéterminé. G est donc Ω-diagnosticable pour $\Sigma_o = \{t_1, t_2, t_4\}$. À noter que $\epsilon(G \times \Omega)$ étant déterministe, $Det(G \times \Omega) = \epsilon(G \times \Omega)$ donc $\epsilon(G \times \Omega)$ correspond également au diagnostiqueur.

5 Conclusion

Nous avons proposé dans cet article un cadre formel permettant de décrire de manière uniforme une classe importante de problèmes de diagnostic, de proposer

des algorithmes généraux pour la vérification de la diagnosticabilité et la construction du diagnostiqueur associé. Les problèmes de diagnostic considérés sont ceux pour lesquels la propriété à diagnostiquer est une propriété d'atteignabilité. La formalisation est basée sur la description séparée du modèle du système et de la propriété à diagnostiquer sous la forme d'un motif de surveillance dont les états finals sont stables. La diagnosticabilité s'exprime alors simplement en termes de langages reconnus par des automates, et les algorithmes de vérification de diagnosticabilité et de construction de diagnostiqueur sont basés sur des constructions standard sur les automates.

Les motifs de surveillance considérés se limitent à des propriétés d'atteignabilité qui ne permettent pas de capturer tous les problèmes de diagnostic de systèmes à événements discrets finis de la littérature. Par exemple certains problèmes de diagnostic de pannes intermittentes [2] nécessiteraient des motifs de surveillance dont les états finals ne seraient pas stables. Diagnostiquer ce type de propriété en observation partielle est plus difficile. Ceci amène les auteurs de [2] d'une part à proposer de nouvelles définitions de diagnosticabilité *ad hoc* qui prennent en compte cette intermittence, d'autre part à restreindre les systèmes de sorte à forcer le retour aux états finals et à exiger une observabilité minimale sur le système. En généralisant nos motifs de surveillance à ce type de propriétés, notre cadre formel peut cependant permettre de reformuler la diagnosticabilité de manière plus générale, d'affiner les restrictions sur les systèmes, d'affiner la vérification de diagnosticabilité et donner des constructions plus simples de diagnostiqueur.

Enfin notre cadre formel peut être étendu à des modèles de systèmes et de surveillance plus généraux. En particulier, nous envisageons l'extension à des modèles de systèmes de transition avec variables. Si le domaine des variables est non-borné, la diagnosticabilité de propriétés d'atteignabilité se ramène à des problèmes d'atteignabilité dans ces modèles et est donc indécidable. La construction de diagnostiqueur devient aussi problématique à cause de l'ϵ-clôture et surtout de la déterminisation. Cependant, de manière similaire à [5], en faisant des restrictions de modèles pour l'ϵ-clôture, en utilisant des heuristiques pour la déterminisation, et une analyse approchée pour l'atteignabilité, il est possible de tester la diagnosticabilité et de construire par des opérations syntaxiques un diagnostiqueur exact manipulant des variables.

Remerciements : nous remercions Sophie Pinchinat pour les discussions préliminaires sur cet article.

Références

[1] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella. Diagnosis of active systems. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 274–278, 1998.

[2] O. Contant, S. Lafortune, and D. Teneketzis. Diagnosis of intermittent faults. *Discrete Event Dynamic Systems : Theory and Applications*, 14(2) :171–202, 2004.

[3] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Event Dynamic System : Theory and Applications*, 10(1/2) :33–86, Janvier 2000.

[4] E. Fabre, A. Benveniste, C. Jard, L. Ricker, and M. Smith. Distributed state reconstruction for discrete event systems. In *IEEE Control and Decision Conference (CDC)*, Sydney, 2000.

[5] B. Jeannet, T. Jéron, V. Rusu, and E. Zinovieva. Symbolic test selection based on approximate analysis. In *11th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'05)Volume 3440 of LNCS*, Edinburgh (Scottland), Avril 2005.

[6] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8) :1318–1321, 2001.

[7] S. Jiang, R. Kumar, and H.E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Robotics and Automation*, 19(2) :310–323, Avril 2003.

[8] Y. Pencolé and M-O. Cordier. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence Journal*, 164(1-2) :121–170, 2005.

[9] L. Rozé and M.-O. Cordier. Diagnosing discrete-event systems : an experiment in telecommunication networks. In *4th International Workshop on Discrete Event Systems*, pages 130–137, 1998.

[10] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9) :1555–1575, 1995.

[11] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology*, 4(2) :105–124, Mars 1996.

[12] T. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Trans. on Automatic Control*, 47(9) :1491–1495, Septembre 2002.