Supervision Patterns in Discrete Event Systems Diagnosis

Jeron Thierry, Marchand Herve, Cordier Marie-Odile, Pinchinat Sophie
IRISA, Campus Universitaire de Beaulieu, 35042 Rennes, France
{firstName.Name}@irisa.fr

January 14, 2006

***Abstract*— not finished**

I. Introduction

not finished

II. Organisation of the paper

not finished

- Definitions and notations : synchronous product, observational-closure, and determisination
- Notion of supervision patterns and Diagnosis Problems;
- Algorithms for the Diagnosis Problems Sophie Complexity Issues?

III. Labelled Transition Systems and Related Notions

A. Labelled Transition Systems

We start first by recalling useful standard notations : We assume given an alphabet Σ, that is a finite set $\{\sigma, \sigma_1, \ldots\}$. The set of finite sequences over Σ is denoted by Σ^*, with ϵ for the empty sequence. In the paper, typical elements of Σ^* are $s, t, u, \ldots$. For each $s, t \in \Sigma^*$ of the form $s = \sigma_1 \ldots \sigma_n$ and $t = \sigma'_1 \ldots \sigma'_m$ $(n, m \in \mathbb{N})$,the *concatenation of s and t* is sill a sequence defined by $s.t = \sigma_1 \ldots \sigma_n \sigma\sigma'_1 \ldots \sigma'_m$. Hence, Σ^* is the least set satisfying $\epsilon \in \Sigma^*$, and $s\sigma \in \Sigma^*$, for all $s \in \Sigma^*$ and all $\sigma \in \Sigma$. The inductive definition of Σ^* is useful for proofs or definitions. As an example, the *length* of $s \in \Sigma^*$, denoted $\|s\|$, is defined by $\|\epsilon\| = 0$ and $\|s\sigma\| = \|s\| + 1$. Also, for $\sigma \in \Sigma$ and $s \in \Sigma^*$, the truthfulness of the assertion "$\sigma \in s$" is defined by "$\sigma \in \epsilon$" is false for all σ, and "$\sigma \in s\sigma'$" is equivalent to "$\sigma = \sigma'$ or $\sigma \in s$".

We now come to the models of systems :

Definition 1 (LTS): An LTS over Σ is defined by a 4-tuple $G = (Q, \Sigma, \rightarrow, q_0)$ where Q is a finite set of states with a distinguished element q_0 called *the initial state*, Σ is the set of *events* of G, $q_0 \in Q$ is the initial state, and $\rightarrow \subseteq Q \times \Sigma \times Q$ is *the partial transition relation.*

In the rest of the section, we assume given an LTS $G = (Q, \Sigma, \rightarrow, q_0)$, we write $q \xrightarrow{\sigma} q'$ for $(q, \sigma, q') \in \rightarrow$, and consider arbitrary sequences by setting : $q \xrightarrow{\epsilon} q$ always holds, and $q \xrightarrow{s\sigma} q'$ whenever $q \xrightarrow{s} q''$ and $q'' \xrightarrow{\sigma} q'$, for some $q'' \in Q$.

G is *deterministic* if whenever $q \xrightarrow{\sigma} q'$ and $q \xrightarrow{\sigma} q''$, then $q' = q''$, for each $q \in Q$ and each $\sigma \in \Sigma$. Hence $\rightarrow$ becomes a (possibly) function.

We set $\Delta_G(q, s) \triangleq \{q' \in Q \mid q \xrightarrow{s} q'\}$. In particular $\Delta_G(q, \epsilon) \triangleq \{q\}$. By abuse of notation, for any subset $\Gamma \subseteq \Sigma^*$, $\Delta_G(q, \Gamma) \triangleq \{q' \in Q \mid q \xrightarrow{s} q' \text{ for some } s \in \Gamma\}$, and for any $Q' \subseteq Q$, $\Delta_G(Q', \Gamma) = \bigcup_{q \in Q'} \Delta_G(q, \Gamma)$.

A subset $Q' \subseteq Q$ is *stable* for $\rightarrow$ whenever $\Delta_G(Q', \Sigma) \subseteq Q'$.

Let $q \xrightarrow{s}$ mean that $q \xrightarrow{s} q'$ for some $q' \in Q$. The *event set* of a state $q \in Q$ is $\Sigma(q) \triangleq \{\sigma \in \Sigma \mid q \xrightarrow{\sigma}\}$. Then G is *complete* whenever $\Sigma(q) = \Sigma$, for each $q \in Q$.

Finally, given $q \in Q$, we let $\mathcal{L}(q) \triangleq \{s \in \Sigma^* \mid q \xrightarrow{s}\}$. The *language generated* by the system G is the set $\mathcal{L}(G) \triangleq \mathcal{L}(q_0)$, which elements are called *executions* of G. Given an execution $s \in \mathcal{L}(G)$, we write

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

for the set of executions that extend s in G.

Finally, a state q is *reachable* if $q \in \Delta_G(q_0, \mathcal{L}(q))$. G is *reachable* if $Q = \Delta_G(q_0, \mathcal{L}(q))$, and it is *alive* if $\mathcal{L}(q) \setminus \epsilon \neq \emptyset$.

As we are interested in diagnosing systems - this will be formalised in the next section -, partial observation plays a central rôle. In this regards, the set of events Σ is partionned into Σ_o and Σ_{uo} (ie $\Sigma = \Sigma_o \cup \Sigma_{uo}$, and $\Sigma_o \cup \Sigma_{uo} = \emptyset$), where Σ_o represents the set of *observable* events - elements of Σ_{uo} are then *unobservable* events. Typical elements of Σ_o^* will be denoted by μ, μ', and typical elements of Σ_{uo}^* are rather written $u, u', \ldots$.

Let $P : \Sigma^* \rightarrow \Sigma_o^*$ be the natural *projection* of executions onto Σ_o^* defined by : $P(\epsilon) = \epsilon$ and $P(s\sigma) = P(s)\sigma$ if $\sigma \in \Sigma_o$, and $P(s)$ otherwise. The projection P simply erases the unobservable events of a sequence. P extends to languages by defining, for $L \subseteq \Sigma^*$, $P(L) = \{P(s) \mid s \in L\}$. Now, the *trace language of* G is

$$Trace(G) \triangleq P(\mathcal{L}(G))$$

It is the set of observations of its executions.

Rapidly in the paper, we will need to distinguish a subset $Q_m \subseteq Q$ to denote final states, in the sense of formal language theory. The notion above are extended in this setting by letting $\mathcal{L}(G, Q_m) = \{\sigma \in \Sigma^* \mid \Delta_G(q_0, \sigma) \subseteq Q_m\}$, and $Trace(G, Q_m) \triangleq P(\mathcal{L}(G, Q_m))$.

From the projection P, we can derive an equivalence relation, called the *observational equivalence* and written $\equiv$, between executions of G. Equivalence classes of $\equiv$ are in a one-to-one correspondance with the traces of G.

Definition 2 (Observational Equivalence, $\equiv$): Let $\equiv \subseteq \mathcal{L}(G) \times \mathcal{L}(G)$ be the binary relation defined by $s \equiv s'$ whenever $P(s) = P(s')$. One easily verifies that $\equiv$ is an equivalence relation, and we take the convention to write $[s]$ for the equivalence class of s.

Given $s \in \mathcal{L}$, s naturally defines the trace $P(s)$. reciprocally, given a trace μ of G, μ uniquely determines the class $[P_G^{-1}(\mu)]$ of executions in G, by letting $P_G^{-1}(\mu) = P^{-1}(\mu) \cap \mathcal{L}(G)$. As a consequence, $\mathcal{L}(G) = P_G^{-1}(Traces(G))$, whereas in general, we only have $\mathcal{L}(G, Q_m) \subseteq P_G^{-1}(Traces(G, Q_m))$, since $P_G^{-1}(Traces(G, Q_m))$ might contain an execution s which does not end up at a state of Q_m.

B. Operations on transition system

We recall three operations over LTSs, all classic ones, inherited from formal language theory . One is binary : it is the classic *Synchronous Product* of LTSs, which aims at defining an LTS which generated language amounts to intersect the generated languages. The two latter are unary operations : the $\Sigma_{uo}^*\Sigma_o$-*closure* and the *Determinzation*.

a) Synchronous product:

Definition 3: Let $G^i = (Q^i, q_0^i, \Sigma, \rightarrow_i)$, $i = 1, 2$, be two LTSs. Their *synchronous product* is $G^1 \times G^2 = (Q^1 \times Q^2, (q_0^1, q_0^2), \Sigma_1 \cup \Sigma_2, \rightarrow)$, where $\rightarrow \subseteq Q^1 \times Q^2$ satisfies $(q'^1, q'^2) \in Q$, $(q^1, q^2) \xrightarrow{\sigma} (q'^1, q'^2)$ whenever $q^1 \xrightarrow{\sigma}_1 q'^1$ and $q^2 \xrightarrow{\sigma}_2 q'^2$.

Moreover, if each G^i $(i = 1, 2)$ is equipped with a set of final states Q_{m_i}, we set the final states of $G^1 \times G^2$ equal to $Q_{m_1} \times Q_{m_2}$. Notice that, by definition, stability component-wise, say of two sets $Q_1 \subseteq Q^1$ and $Q_2 \subseteq Q^2$, implies stability in their product, that is $Q_1 \times Q_2$ is stable in $G^1 \times G^2$.

As announced earlier, $\mathcal{L}(G^1 \times G^2) = \mathcal{L}(G^1) \cap \mathcal{L}(G^2)$ and $\mathcal{L}(G, Q_m) = \mathcal{L}(G^1, Q_{m_1}) \cap \mathcal{L}(G^2, Q_{m_2})$.

Final, when we distinguish a subset Σ_o of the global alphabet $\Sigma_1 \cup \Sigma_2$ the global alphabet Σ, and if we consider the traces, the picture is less ideal than for executions : in general we only have $Trace(G^1 \times G^2) \subseteq Trace(G^1) \cap Trace(G^2)$ et $Trace(G^1 \times G^2, Q_m) \subseteq Trace(G^1, Q_{m_1}) \cap Trace(G^2, Q_{m_2})$.

b) Observational-closure: The observational-closure operates on an LTS in order to eliminate unobservable events while preserving traces. Slightly different from to the classic ϵ-transitive closure in formal language theory, only sequences belonging to $\Sigma_{uo}^*\Sigma_o$ are shorten. Formally,

Definition 4 (Observational-closure): Assume an LTS $G = (Q, \Sigma, \rightarrow, q_0)$ and the partition $\Sigma = \Sigma_{uo} \cup \Sigma_o$. The observational-closure of G *is $Obs(G) = (Q, \Sigma_o, \rightarrow_\epsilon, q_0)$, where $q \xrightarrow{\sigma}_\epsilon q'$ whenever there exist $u \in \Sigma_{uo}^*$ and $\sigma \in \Sigma_o$ s.t. $q \xrightarrow{u\sigma} q'$.*

Notice that by definition $\mathcal{L}(Obs(G)) = Trace(Obs(G)) = Trace(G)$, hence preserving traces.

c) Determinisation: Given an LTS G, the Determinisation of G consists in computing a deterministic LTS over Σ_o which traces are still $Trace(G)$. It is standard in classic formal language theory :

Definition 5 (Determinization): Given an LTS $G = (Q, \Sigma, \rightarrow, q_0)$ and a partition $\Sigma = \Sigma_{uo} \cup \Sigma_o$, the determinization of G *is $Det(G) = (\mathcal{X}, \Sigma_o, \rightarrow_d, X_0)$ where $\mathcal{X} \subseteq \mathcal{P}(Q)$, $X_0 = \{q_o\}$ and $\xrightarrow{\sigma}_d = \{(X, \sigma, \Delta_G(X, \Sigma_{uo}^*.\sigma) \mid X, X' \in \mathcal{X}$ and $\sigma \in \Sigma_o\}$.*

One can establish that $Trace(G) = \mathcal{L}(Det(G))$. Note that if $\mu \in Trace(G)$ and $\mu \neq \epsilon$, we have

$$\Delta_{Det(G)}(x_0, \mu) = \{\Delta_G(q_0, s) \mid [s] = \mu \text{ and } s \in \cap\Sigma^*.\Sigma_o\} \quad (1)$$

Sophie I still think that the above is usrless and confusing

An alternative definition of $Det(G)$ can be given by using the Obs-closure of G.

Exemple 1: in order to illustrate the previous concepts, let us consider the following LTS G given in Figure 1, with $\Sigma = \{a, b, d, f, t\}$ and $\Sigma_{uo} = \{f\}$.

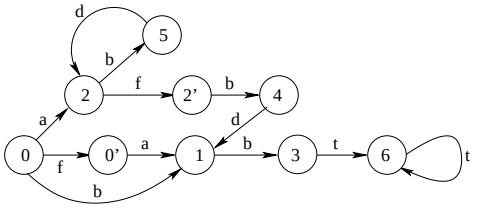


Fig. 1. G

Then $Obs(G)$ and $Det(G)$ are given by the LTS given in Figure 2.

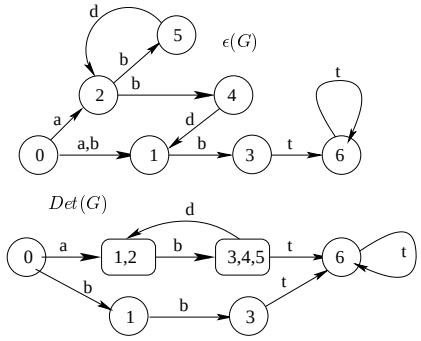


Fig. 2. $Obs(G)$ et $Det(G)$

Sophie We should recall here the algorithmic cost of these operations.

IV. Supervision Patterns and the Diagnosis Problem

A. Supervision Patterns

Supervision patterns are represented by particular LTSs :

Definition 6: A supervision pattern *is a structure* $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega}, Q_F)$*, where* $(Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega})$ *is deterministic and complete LTS, and* $Q_F \subseteq Q$ *is a distinguished stable subset of states with* $q_{0_\Omega} \notin Q_F$*.*

In the sequel, by abuse of notations and and because this is the only use of supervisory pattern, we take the convention to simply write $\mathcal{L}(\Omega)$ instead of $\mathcal{L}(\Omega)$.

In the next subsection we will give many classic problems in diagnosis which are rephrased in terms of supervision patterns. Notice that Definition 6 requires that $q_{0_\Omega} \notin Q_F$; this is very natural. Indeed, a supervision pattern is not design to declare a fault *a priori* while the system has not made any transition yet. Henceforth, we do not consider that unexpected behaviour (membership in Q_F) can occur from the very initial state of a system.

Now, given Ω a supervision pattern and G an LTS, we make precise what G being Ω-diagnosable means :

Definition 7: An LTS G is Ω*-diagnosable* whenever

$$\begin{array}{l}\exists n \in \mathbb{N}, \forall s \in \mathcal{L}(\Omega) \cap \mathcal{L}(G), \forall t \in \mathcal{L}(G)/s, \\ \text{if } \|P(t)\| \geq n \text{ then } [s.t] \subseteq \mathcal{L}(\Omega),\end{array} \quad (2)$$

that is, any execution s' which is $\Sigma_{uo}^*\Sigma_o$-equivalent to $s.t$ also belongs to $\mathcal{L}(\Omega)$.

Sophie Ce qui suit est sans doute à déplacer pour que cela apparaisse après les exemple qu'Hervé intégrera et qui sont pour le moment commentés dans mon source.

The Ω-diagnosability subsumes the f-diagnosability [5], [6]. f-diagnosability is meaningful only under the assumption that any infinite sequence of G necessarily contains an observable event. Writing $\texttt{endingby}(f)$ for the set of executions of G that end up with f, rather written $\Psi(f)$ by the authors, we says that G if f-diagnosable whenever

$$\begin{array}{l}\exists N \in \mathbb{N}, \forall s \in \texttt{endingby}(f), \forall t \in \mathcal{L}(G)/s, \\ \text{if } \|t\| \geq N, \text{ then } \forall u \in [s.t], f \in u\end{array} \quad (3)$$

We now establish that :

Proposition 1: Let G be an LTS. Then if G is f-diagnosable then G is Ω-diagnosable. If moreover any infinite sequence of G contains an observable event, the reciprocal also holds.

Proof We first make the following remarks :

(a) $f \in u$ is equivalent to $u \in \mathcal{L}(\Omega)$;

(b) $s \in \texttt{endingby}(f)$ implies $s \in \mathcal{L}(\Omega)$;

Assume G is f-diagnosable; in particular, consider a natural number N satisfying Property(3). We prove that Ω-diagnosability holds for the same natural number N : consider $s \in \mathcal{L}(\Omega)$ and let $t \in \mathcal{L}(G)/s$ with $\|P(t)\| \geq N$; note that therefore $\|t\| \geq N$. It is easy show that s decomposes into $s = s's''$ where $s' \in \texttt{endingby}(f)$, with additionally $s''.t \in \mathcal{L}(G)/s'$. Now, by construction $\|s''.t\| \geq N$, which, by (3), entails that any $u \in [s'.s''.t]$ is also in $\mathcal{L}(\Omega, Q_F)$, which concludes.

Reciprocally, assume G is Ω-diagnosable. We still use the remarks (a) and (b) above plus the fact that by letting m be the length of the longest unobservable sequence in G (which exists by assumption), $\|P(t)\| \geq n$ implies $\|t\| \geq n * m$. Now, it is easy to establish that G is f-diagnosable by taking $N = n * m$. ◇

B. Plethora of Examples

d) occurence of multiple faults: Let f_1 and f_2 be two faults that may occur in the system. Let Ω_{f_1} and Ω_{f_2} the corresponding supervision patterns of f_1 and f_2 as described in Figure **??**.

If one want to diagnose the occurrence of the two faults, then it is sufficient to consider the supervision pattern, described in figure 3, obtained by performing the parallel product between Ω_{f_1} and Ω_{f_2}, with $Q_F = (F_1, F_2)$ as final state. This new supervision pattern accepts the sequences of $\mathcal{L}(\Omega_{f_1}, F_1) \cap \mathcal{L}(\Omega_{f_2}, F_2)$.

In general, when one have to diagnose a set of faults $\{f_1, \cdots, f_l\}$, it is necessary to perform the product between the atomic supervision patterns Ω_{f_i}. Thus, the size of the global supervision pattern is in $\mathcal{O}(2^l)$. If one wants to diagnose one fault among the set of faults $\{f_1, \cdots, f_l\}$, then the global supervision pattern will be given as the FSM that accepts the union of the languages accepted by the FSM Ω_{f_i}. The size of this FSM is in $\mathcal{O}(l)$.

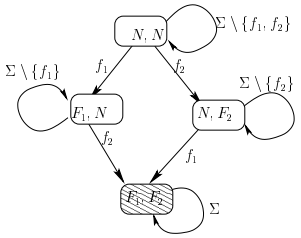


Fig. 3. Supervision pattern for two faults

If the diagnosis that has to be performed concern the occurrences of different faults in a precise order, for example, f_2 after f_1, the sequences that have to be recognized by the supervision pattern is

$$(\Sigma \setminus \{f_1\})^*.f_1.(\Sigma \setminus \{f_2\})^*.f_2.\Sigma^*,$$

which corresponds (more or less) to the concatenation of the two languages $\mathcal{L}(\Omega_{f_1}, F_1).\mathcal{L}(\Omega_{f_2}, F))$ as described by the supervision pattern given in Figure 4.

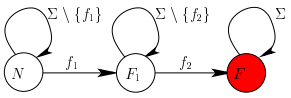


Fig. 4. Pannes en cascade

e) multiples Occurrence of the same fault: Another interesting problem is to diagnose the multiple occurrences of the same fault event f. For this kind of problem, one possible supervision pattern is given in Figure 5. The sequences accepted by this FSM are given by $\mathcal{L}(\mathcal{O}_f, F)^k$. This allows to diagnose that the fault f occurs at least k times in the system. This corresponds to the k-diagnosability of [4].

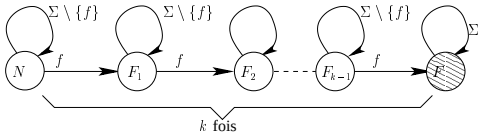


Fig. 5. k occurrences de la panne f

One can also consider has faulty set of final states the sets $(\{F_i, \cdots, F_{k-1}, F\})_{k' \leq i \leq k}$. A diagnoser computed with respect to one of this sets diagnose that the fault f occurs at least i times (with $i \geq k'$). Now we can also consider the union of the verdict of all these diagnosers. We thus diagnose that a fault occurred at least between k' and k times. With $k' = 1$, this actually corresponds to the $[1-k]$-diagnosability of [4].

f) Intermittent Fault: So far, we have considered permanent faults. However, there exist numerous systems in which fault are intermittent (i.e. the effect of the fault can be repaired). We here assume that reparation of a fault f is encoded by means of an event r (see [1] for details). The supervision pattern given in Figure 6 describes the fact that a fault occurred in a system and that the system has been further repaired (however, when this pattern is recognized, the system may be faulty). This corresponds to the *I-diagnosability* in [1].

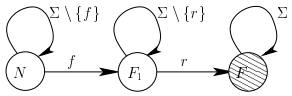


Fig. 6. Intermittent fault with reparation

Remarque 1: By considering the states F_1 and F as final, we just diagnose the fact that a fault occurred in the system. This is what is called O-diagnosability in [1].

Another example is given in Figure 7, where the supervision pattern allows to detect that the fault occurred twice without having being repaired.

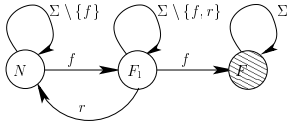


Fig. 7. Intermittent fault with reparation (II)

g) Another example of supervision: The example given in Figure 8 described a system with internal events loops. In this example, we simply model the movement of a person in a building composed of an *office* (I), a *library* (B), a *reception* (A) and a *coffee-shop* (C). The doors from one part of the building to another can be taken in only one direction. Some are secured by access-cards (allowing the observation). We assume that there exist access-cards for the

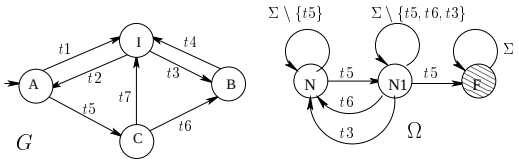


Fig. 8. G and the corresponding supervision pattern Ω

doors t_1, t_2 and t_4, but some of them may be deactivated (i.e. the access-card is usefulness and there is now way to observe the fact that one person cross the door). We consider the supervision pattern given in Ω (Figure 8) which expresses the fact that going twice to coffee-shop without going to the library is a behavior that has to be supervised. We will come back to this example further in the paper.

C. The Diagnosis Problems

The *Diagnosis problems* assume given a system G and a supervisory pattern Ω. It consists in two problems : the first one is the *Diagnosability Problem* where the Ω-diagnosability of G is addressed (ie whether Property (2) holds). The second one is the *Diagnosis Synthesis Problem* which aims at computing a boolean function $\texttt{In}\mathcal{L}(\Omega)$ of $\mathcal{L}(G)$ approximating the membership of an execution in $\mathcal{L}(\Omega)$. Indeed, as G is only partially observed, we do not expect in general the function $\texttt{In}\mathcal{L}(\Omega)$ to be accurate (we know this is undecidable by [7], but we require the following properties :

- (Verdict Correctness) For any $s \in \mathcal{L}(G)$, if $\texttt{In}\mathcal{L}(\Omega)(s)$ then $s \in \mathcal{L}(\Omega)$;
- (Bounded Completeness) There exists $n \in \mathbb{N}$, the bound, such that for any $s \in \mathcal{L}(G)$, if $s \in \mathcal{L}(\Omega)$, then there exists some extension $s' \in \mathcal{L}(G)$ of s with $\|s'\| - \|s\| \leq n$, such that $\texttt{In}\mathcal{L}(\Omega)(s')$;
- (Σ_o-Consistency) Given two $\Sigma_{uo}^* \Sigma_o$-equivalent executions of G, say s and s', we have $\texttt{In}\mathcal{L}(\Omega)(s)$ if and only if $\texttt{In}\mathcal{L}(\Omega)(s')$.

Notice that by the above, $\neg\texttt{In}\mathcal{L}(\Omega)(s)$ does not mean $s \notin \mathcal{L}(\Omega)$, whereas $\texttt{In}\mathcal{L}(\Omega)(s)$ indeed guaranties that $s \in \mathcal{L}(\Omega)$.

V. Algorithms for the Diagnosis Problems

We now discuss algorithms for the the Diagnosis Problems. In a first stage, we propose a reduction of the Ω-diagnosability to a state-based problem, called a State-Diagnosis Problem. From this reduction, we propose a

general sound algorithm, making the Ω-diagnosability decidable. Next, and still relying on the former reduction, we describe an algorithm to synthesise a labelling of $\mathcal{L}(G)$ which achieves the diagnosis function expected in the Diagnosis Synthesis Problem.

A. Supervision Patterns and State Diagnosis

Definition 8: We assume given an LTS $G = (Q, \Sigma, q_0, \rightarrow)$ where Σ is partitionned into Σ_o and Σ_{uo}, such that G is deterministic, reachable, alive and s.t. any terminal strongly connected component of G contains an observable event Moreover, we assume given a distinguished subset of states $Q_{\text{F}} \subseteq Q$, we say that G is Q_P*-diagnosable* whenever

$$\exists n \in \mathbb{N},\ \forall s \in \mathcal{L}(G, Q_{\text{P}}), \forall t \in \mathcal{L}(G)/s,$$
$$\|P(t)\| \geq n \ \Rightarrow [s.t] \subseteq \mathcal{L}(G, Q_{\text{P}})$$

From Definitions **??** and 8, one easily established that : G is Ω-diagnosable iff $G \times \Omega$ est $Q \times Q_F$-diagnosable

Sophie la suite jusqu'à est à traduire Intuitivement, un systeme G est Q_{P}-diagnosticable si et seulement si apres une sequence s du systeme reconnue par Q_{P}, ($s \in \mathcal{L}(G, Q_{\text{P}})$), si on observe suffisamment d'evenements ($t \in \mathcal{L}(G)/s, \|P(t)\| \geq n$), alors toutes les sequences compatibles avec l'observation ($P_G^{-1}(P(s.t))$) sont aussi reconnues par Q_{P} (appartiennent à $\mathcal{L}(G, Q_{\text{P}})$).

L'hypothese que G n'a pas de composante fortement connexe terminale d'events inobservables permet d'eviter les cas où une event inobservable ferait transiter dans Q_{P} sans qu'aucune observation ne suive et ne permette de le diagnostiquer. Combine avec la vivacite, ceci implique que $Det(G)$ est aussi vivant.

Remarque 2: Si le systeme G ne possede aucun cycle d'events inobservables, (comme suppose dans [5], [6]) il est equivalent de faire porter la borne n sur la longueur des sequences ou sur la longueur de l'observation. Notre definition est donc legerement plus generale que celle de [5], [6]. $\diamond$

Le resultat suivant montre qu'on peut ramener le probleme de la Ω-diagnosticabilite defini dans la section precedente à celui de la Q_{P}-diagnosticabilite:

B. Construction du diagnostiqueur.

Sophie je n'ai pas encore expliqué les fameux $s \times s'$. Je vous le soumets ce soir.

Etant donne le caractere inobservable d'une partie des evenements du systeme, le diagnostiqueur doit travailler sur une estimation de son etat courant (caracterisee par un ensemble d'etats possibles, dans lesquels le systeme peut avoir evolue apres une trace donnee). Le diagnostiqueur est donc simplement un observateur externe au systeme qui, sur la base de ses observations, doit estimer si les sequences compatibles avec cette observation sont reconnues ou non par Q_{P}. Il est construit à partir de $Det(G)$ comme suit :

Definition 9: Soit un systeme à diagnostiquer (G, Q_{P}) avec $G = (Q, \Sigma, \rightarrow, q_0)$ et $Q_{\text{P}} \subseteq Q$.

Le diagnostiqueur de (G, Q_{P}) est le LTS determinise $G_d = Det(G) = (\mathcal{X}, \Sigma_o, x_o, \rightarrow_d)$, muni de la fonction DIAG*: $\mathcal{X} \longrightarrow \{P, N, In\}$ definie par*

$$\text{DIAG}(x) = \begin{cases} P & si \quad x \subseteq Q_{\text{P}} \\ N & si \quad x \cap Q_{\text{P}} = \emptyset \\ In & autrement \end{cases} \tag{4}$$

Un etat $x \in \mathcal{X}$ 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 sequences de G compatibles avec l'observation μ et se terminant par une observation. Intuitivement, le diagnostic P est donc emis quand toutes les sequences de $Comp(\mu)$ sont acceptees dans Q_{P}. Comme Q_{P} est stable, toute prolongation par des events inobservables ne change pas l'acceptation en Q_{P}, donc ceci est equivalent au fait que toutes les sequences de $P_G^{-1}(\mu)$ sont reconnues par Q_{P}.

Sophie Stability requirement : what about the stability according to unobservable events, so that observable ones can take you out of Q_{P}?

Le diagnostic N est emis quand aucune des sequences 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 $Q \setminus Q_{\text{P}}$ par des events inobservables. Le diagnostic In est emis dans tous les autres cas, i.e. les cas où certaines sequences de $Comp(\mu)$ sont acceptees et d'autres non.

Sophie at this stage I am not sure that the notation $Comp(\mu)$ is worth. First it is not used in the proposition below, this can be arranged, but moreover it does not seem to be used in what follows... Finaly, it is not defined for the empty sequence.

La proposition suivante formalise ces remarques :

Proposition 2: $\forall \mu \in \mathcal{L}(G_d)$,

$$\Delta_{G_d}(x_0, \mu) \subseteq Q_{\text{P}} \iff P_G^{-1}(\mu) \subseteq \mathcal{L}(G, Q_{\text{P}}) \tag{5}$$

$$\begin{aligned} &\Delta_{G_d}(x_0, \mu) \cap Q_{\text{P}} = \emptyset \\ &\qquad \iff P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o \cap \mathcal{L}(G, Q_{\text{P}}) = \emptyset \end{aligned} \tag{6}$$

Proof Sophie If $\mu = \epsilon$, then (5) clearly holds as on the one hand, from the assumption that $q_0 \notin Q_{\text{P}}$ we get $\epsilon \notin \mathcal{L}(G, Q_{\text{P}})$ although $\epsilon \in P_G^{-1}(\epsilon)$, and on the other hand, from $q_0 \in x_0$ and $x_0 = \Delta_{G_d}(x_0, \epsilon)$, we get $\Delta_{G_d}(x_0, \epsilon) \not\subseteq Q_{\text{P}}$.

Assume now that $\mu \neq \epsilon$, and that $\Delta_{G_d}(x_0, \mu) \subseteq Q_{\text{P}}$. Consider $s \in P_G^{-1}(\mu)$. If s ends by an observable event, then by definition of $\Delta_{G_d}(x_0, \mu)$, $\Delta_G(q_0, s) \in Q_{\text{P}}$ and we are done. Otherwise s decomposes into $s = t.t'$ where t ends with an observable event. Then by the previous reasonning, $\Delta_G(q_0, t) \in Q_{\text{P}}$. The set Q_{P} being stable, we also have $\Delta_G(q_0, t.t') \in Q_{\text{P}}$. Reciprocally, if $P_G^{-1}(\mu) \subseteq \mathcal{L}(G, Q_{\text{P}})$, then because any state q in $\Delta_{G_d}(x_0, \mu)$ is by definition reachable by a sequence of $P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o$, we have $q \in Q_{\text{P}}$.

For (6), if $\mu = \epsilon$, since both expressions $\Delta_{G_d}(x_0, \epsilon) \cap Q_{\text{P}}$ and $P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o$ are empty, we are done. Otherwise, by (1), $q \in \Delta_{G_d}(x_0, \mu)$ if and only if $q = \Delta_G(q_0, s_q)$ for some $s_q \in P_G^{-1}(\mu) \cap \Sigma^*.\Sigma_o$. Henceforth $s_q \in \mathcal{L}(G, Q_{\text{P}})$ if and only if $q \in Q_{\text{P}}$. Which concludes. ◇

Il reste à montrer que si le systeme est diagnosticable, le diagnostiqueur construit en Definition 9 est correct. Ceci est etabli par la Proposition 3 qui dit que si une sequence du systeme penetre Q_{P}, le diagnostiqueur construit nous l'indiquera en temps fini (i.e. apres l'occurrence d'un nombre fini d'observations).

Proposition 3: Soit $G = (Q, \Sigma, \rightarrow, q_0, Q_P)$ un systeme à diagnostiquer et son diagnostiqueur $G_d = Det(G)$ muni de sa fonction DIAG. *Si G est Q_P-diagnosticable alors*

$$\begin{array}{l} \exists n \in \mathbb{N}, \forall s \in \mathcal{L}(G, Q_P), \forall t \in \mathcal{L}(G)/s, \\ \|P(t)\| \geq n \Rightarrow \text{DIAG}(\Delta_{G_d}(x_o, P(s.t)) = P \end{array} \quad (7)$$

Proof Rappelons que la Q_{P}-diagnosticabilite est definie par,

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

En appliquant la proposition 2 à $P(s.t)$ on a $P_G^{-1}(P(s.t)) \subseteq \mathcal{L}(G, Q_{\text{P}})$ si et seulement si $\Delta_{G_d}(x_0, P(s.t)) \subseteq Q_{\text{P}}$ ce qui equivaut à $Diag(\Delta_{G_d}(x_0, P(s.t))) = P$, ce qui termine la preuve. ◇

Exemple 2: Pour illustrer la definition 9, considerons le LTS G (adapte de [5]) represente en Figure 9. Supposons que dans G, $Q_P = \{0', 1, 2', 3, 4, 6\}$ ce qui correspond aux etats atteignables apres l'occurrence de la panne modelisee par l'evenement f. Suivant la definition 9, les etats $\{1, 2\}$ et $\{3, 4, 5\}$ de $\mathcal{X}$ sont indetermines (i.e. etiquetees In) pour le diagnostic, alors que $\{6\}$ indique que le systeme est sûrement en panne.

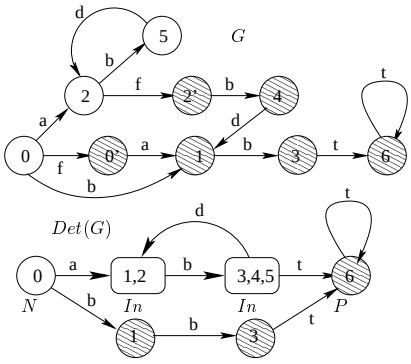


Fig. 9. G et son diagnostiqueur associe

On peut toutefois constater que le diagnostiqueur seul ne permet pas de conclure que le systeme est diagnosticable ou non. Il existe un cycle d'etats indetermines entre $\{1, 2\}$ et $\{3, 4, 5\}$[1], qui laisse à penser que si l'on observe une sequence arbitrairement longue $a(bd)^n, n \in \mathbb{N}$, alors on ne saura jamais pas si le systeme est en panne ou non. Ce systeme est pourtant diagnosticable. En effet des que le systeme evolue dans un etat panne ($0'$ ou $2'$), alors l'evenement t sera observe apres un nombre d'evenements borne par 4. L'observation de t indique donc de maniere certaine que le systeme est passe par f. ◇

Exemple 3: A contrario, considerons l'exemple suivant donne en Figure 10. Ce systeme n'est pas diagnosticable

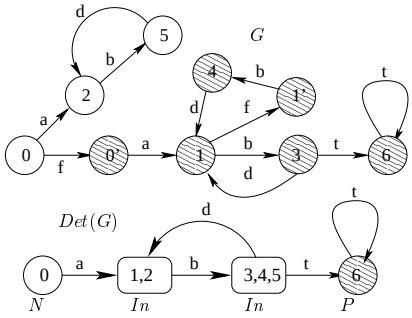


Fig. 10. G et son diagnostiqueur associe

car le cycle indetermine dans le diagnostiqueur correspond cette fois-ci à de vrais cycles indetermines dans le systeme, 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 systeme est diagnosticable. Ceci a conduit à proposer des algorithmes de verification de diagnosticabilite qui se basent, soit sur le diagnostiqueur et le systeme [6] soit sur $\epsilon(G)$ [3]. ◇

C. Verification de la diagnosticabilite.

Un des aspects importants de la diagnosticabilite consiste à savoir *a priori* verifier qu'un systeme est diagnosticable ou non. Les deux exemples precedents nous ont montre que le diagnostiqueur seul ne permet pas cette verification (la presence d'un cycle indetermine (i.e. etiquete In) dans celui-ci n'implique pas forcement que le systeme soit non-diagnosticable).

Rappelons que les hypotheses faites sur le systeme impliquent que $Det(G)$ est vivant. Cette hypothese nous assure que quelque soit le comportement du systeme, il y aura toujours une observation possible. La methode de verification de diagnosticabilite de [8], [3] peut être adaptee à la diagnosticabilite de l'entree dans un ensemble d'etats stable. La preuve, adaptee de [3], est toutefois plus directe. L'idee intuitive de la verification est que le systeme n'est pas diagnosticable lorsqu'il existe deux sequences de longueurs arbitrairement longues, equivalentes du point de vue de l'observation, l'une traversant des etats de Q_{P}, l'autre non. Afin de pouvoir identifier de tels couples de sequences, nous introduisons le LTS suivant.

Soit (G, Q_P) un systeme à diagnostiquer et $\epsilon(G) = (Q_\epsilon, \Sigma_o, q_0, \rightarrow_\epsilon)$ l'ϵ-clôture de G. On note G_o le LTS defini

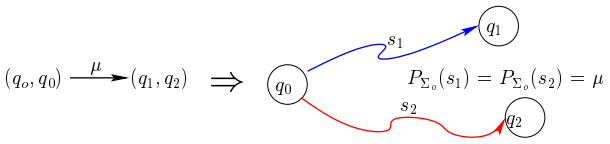
[1] i.e. $\text{DIAG}(\{1, 2\}) = \text{DIAG}(\{3, 4, 5\}) = In$

par

$$\begin{aligned} G_o &= \epsilon(G) \times \epsilon(G) \\ &= (\mathcal{X}_o, \Sigma_o, (q_o, q_o), \rightarrow_o), \text{ avec } \mathcal{X}_o = Q_\epsilon \times Q_\epsilon \end{aligned} \quad (8)$$

G_o va permettre d'identifier des couples de sequences equivalentes menant dans des etats differents du systeme. En effet,

Lemme 1: Soit G_o tel que defini en 8. 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 sequences 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 $P(s_1) = P(s_2) = \mu$.*



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 etiquete par μ', alors pour tout l, la sequence $\mu.\mu'^l$ ne permet pas de diagnostiquer l'entree dans Q_P. Nous allons montrer que l'absence de tels cycles est une condition necessaire et suffisante à la diagnosticabilite.

Definition 10: Soit $x = (q_1, q_2)$ un etat de G_o. On dira que x est indetermine *si $q_1 \notin Q_P$ et $q_2 \in Q_P$ ou si $q_1 \in Q_P$ et $q_2 \notin Q_P$. Un cycle d'etats Q_P-indetermines 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 indetermine.

La proposition suivante etablit une condition necessaire et suffisante de diagnosticabilite et donne un moyen algorithmique de la verifier.

Proposition 4: G est Q_P-diagnosticable ssi il n'existe pas de cycle d'etats Q_P-indetermines dans G_o.

Proof Sophie

$\Rightarrow$) We prove the modus tolens by considering the case where there indeed exist a cycle of undertemined states in $G_o \times G_o$. Sophie (why is it reachable?)

Let $(q_0, q_0) \rightarrow_o \ldots \rightarrow_o (q, q') \ldots \rightarrow_o (q, q')$ be an execution reaching this cycle, and let $s \times s'$ be a sequence of synchronized events that underly a sub-sequence $(q_0, q_0) \rightarrow_o \ldots \rightarrow_o (q, q')$, and $t \times t'$ be the the sequence underlying the cycle from (q, q') back to itself. Without loss of generality we can assume that $q \in Q_\text{p}$ and $q' \notin Q_\text{p}$; from the stability of Q_p, we can assert that any state (r, r') along the sequence $s.t \times s'.t'$ is such that $r' \notin Q_\text{p}$. Now there are arbitrarily long sequences, namely $s.(t)^l$ and $s'.(t'')^l$ (with $l \in \mathbb{N}$), which disagrre on $\mathcal{L}(G, Q_\text{p})$ membership.

$\Leftarrow$) Assume there is no infinite sequence of undetermined states in G_0 and write N for the length of a longuest sequence of undetermined states. We prove the Q_P-diagnosable of G for theatural number $N + 1$.

Let $s \in \mathcal{L}(G, Q_\text{P})$, let $t \in \mathcal{L}(G/s)$ with $\|P(t)\| \geq n$, and let $u \in P_G^{-1}(P(s.t))$. We show that $u \in \mathcal{L}(G, Q_\text{P})$.

Let $(q_0, q_0) \rightarrow_o (q_1, q_1') \rightarrow_o (q_2, q_2') \ldots \rightarrow_o (q_k, q_k')$ be the sequence consisting of the states that are traversed by executing $s.t \times u$ in $G_0 \times G_0$. Let u' denote the suffixe of u which is synchronized with t, along this execution.

Because $s \in \mathcal{L}(G, Q_\text{P})$, there must exist a state (q_i, q_i') with $q_j \in Q_\text{P}$ for all $i \leq j \leq k$. Now, if $q_i' \in Q_\text{P}$, then $q_k' \in Q_\text{P}$ as Q_P is stable, which entails $u \in \mathcal{L}(G, Q_\text{P})$. Otherwise, the sub-sequence from (q_i, q_i') to (q_k, q_k') is the one traversed while executing $t \times u'$. This sub-sequence is strictly longer than N since $\|t\| \geq \|P(t)\| \geq N + 1$. By assumption on N there must exist some state determined states (q_i, q_j') with $i \leq j \leq k$. Now from $q_j \in Q_\text{P}$ we get $q_j' \in Q_\text{P}$, which shows that $u \in \mathcal{L}(G, Q_\text{P})$.

Exemple 4: Pour illustrer ce dernier point, considerons l'exemple 2. On suppose que $Q_P = \{0', 1, 2', 3, 4, 6\}$.

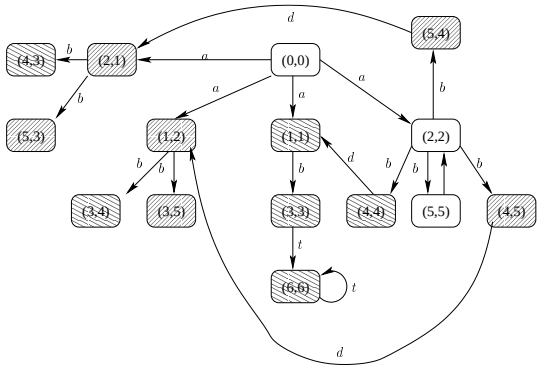


Fig. 11. G_o pour le LTS de l'exemple 2 de la Figure 9

Les tuples $\{(1,2), (2,1), (3,5), (5,3), (4,5), (5,4)\}$, dans G_o, sont indetermines. Il est facile de voir qu'il n'y a aucun cycle d'etats indetermines. Par consequent G est Q_P-diagnosticable. ◇

Remarque 3: On peut noter que G_d et G_o sont independants de l'ensemble des etats modelisant les etats de pannes. Changer cet ensemble ne necessite pas de recalculer ces deux LTS. ◇

h) Supervision example: Let us come back to the example given in Figure 8. Following the different steps described in the previous section, the product between $G \times \Omega$ is used to label the states of G with respect to the supervision pattern Ω. The corresponding LTS is described in Figure 12. Let us first assume that only the access-cards, corresponding to the events t_1, t_2 are activated and thus observable, (i.e. $\Sigma_o = \{t_1, t_2\}$). The observable system (i.e. $\epsilon(G \times \Omega)$) is given in Figure 13. It is easy to check that the LTS $G_o = \epsilon(G \times \Omega) \times \epsilon(G \times \Omega)$ (not represented here) has an undetermined reachable:

$$\begin{aligned} &((A,N),(A,N)) \xrightarrow{t_2}_o ((A,N),(A,N1)) \xrightarrow{t_2}_o \\ &\quad \cdots ((A,N),(A,F)) \xrightarrow{t_2} ((A,N),(A,F)) \end{aligned}$$

[2] Si G etait indeterministe, ce resultat ne serait plus vrai.

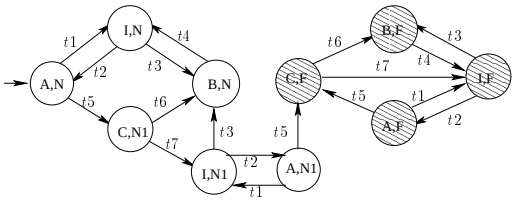


Fig. 12. $G \times \Omega$

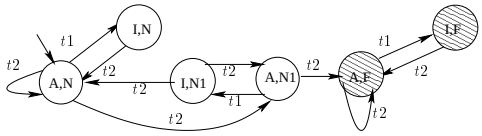


Fig. 13. $\epsilon(G \times \Omega)$ for $\Sigma_o = \{t_1, t_2\}$

thus G is not Ω-diagnosable w.r.t. the set of observable $\Sigma_o = \{t_1, t_2\}$.

However, if the access-card t_4 is activated (i.e. $\Sigma_o = \{t_1, t_2, t_4\}$), the observable system $\epsilon(G \times \Omega)$ is then given by the LTS represented in Figure 14.

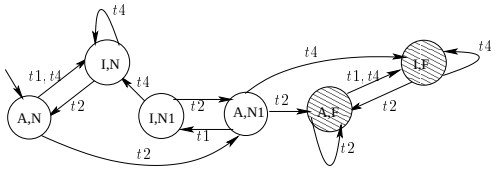


Fig. 14. $\epsilon(G \times \Omega)$ for $\Sigma_o = \{t_1, t_2, t_4\}$

This system is deterministic. Thus $G_o = \epsilon(G \times \Omega) \| \epsilon(G \times \Omega) = \epsilon(G \times \Omega)$ and G_o does not have undetermined cycle. Thus G is Ω-diagnosable for $\Sigma_o = \{t_1, t_2, t_4\}$. Note that we also have that $Det(G \times \Omega) = \epsilon(G \times \Omega)$. Thus $\epsilon(G \times \Omega)$ actually corresponds to the diagnoser.

VI. Conclusion

Nous avons propose dans cet article un cadre formel permettant de decrire de maniere uniforme une classe importante de problemes de diagnostic, de proposer des algorithmes generaux pour la verification de la diagnosticabilite et la construction du diagnostiqueur associe. Les problemes de diagnostic consideres sont ceux pour lesquels la propriete à diagnostiquer est une propriete d'atteignabilite. La formalisation est basee sur la description separee du modele du systeme et de la propriete à diagnostiquer sous la forme d'un motif de surveillance dont les etats finals sont stables. La diagnosticabilite s'exprime alors simplement en termes de langages reconnus par des automates, et les algorithmes de verification de diagnosticabilite et de construction de diagnostiqueur sont bases sur des constructions standard sur les automates.

Les motifs de surveillance consideres se limitent à des proprietes d'atteignabilite qui ne permettent pas de capturer tous les problemes de diagnostic de systemes à evenements discrets finis de la litterature. Par exemple certains problemes de diagnostic de pannes intermittentes [1] necessiteraient des motifs de surveillance dont les etats finals ne seraient pas stables. Diagnostiquer ce type de propriete en observation partielle est plus difficile. Ceci amene les auteurs de [1] d'une part à proposer de nouvelles definitions de diagnosticabilite *ad hoc* qui prennent en compte cette intermittence, d'autre part à restreindre les systemes de sorte à forcer le retour aux etats finals et à exiger une observabilite minimale sur le systeme. En generalisant nos motifs de surveillance à ce type de proprietes, notre cadre formel peut cependant permettre de reformuler la diagnosticabilite de maniere plus generale, d'affiner les restrictions sur les systemes, d'affiner la verification de diagnosticabilite et donner des constructions plus simples de diagnostiqueur.

Enfin notre cadre formel peut être etendu à des modeles de systemes et de surveillance plus generaux. En particulier, nous envisageons l'extension à des modeles de systemes de transition avec variables. Si le domaine des variables est non-borne, la diagnosticabilite de proprietes d'atteignabilite se ramene à des problemes d'atteignabilite dans ces modeles et est donc indecidable. La construction de diagnostiqueur devient aussi problematique à cause de l'ϵ-clôture et surtout de la determinisation. Cependant, de maniere similaire à [2], en faisant des restrictions de modeles pour l'ϵ-clôture, en utilisant des heuristiques pour la determinisation, et une analyse approchee pour l'atteignabilite, il est possible de tester la diagnosticabilite et de construire par des operations syntaxiques un diagnostiqueur exact manipulant des variables.

i) Remerciements :: nous remercions Sophie Pinchinat pour les discussions preliminaires sur cet article.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Stavros Tripakis. Undecidable problems of decentralized observation and control on regular languages. *Inf. Process. Lett.*, 90(1):21–28, 2004.

[8] 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.